Tutorial PM-2

Time Series Similarity Measures

Dimitrios Gunopulos (UC, Riverside)
Gautam Das (Microsoft Research)

Time Series Similarity Measures

Dimitrios Gunopulos
University of California, Riverside
dg@cs.ucr.edu

and

Gautam Das
Microsoft Research
gautamd@microsoft.com

Time Series Databases

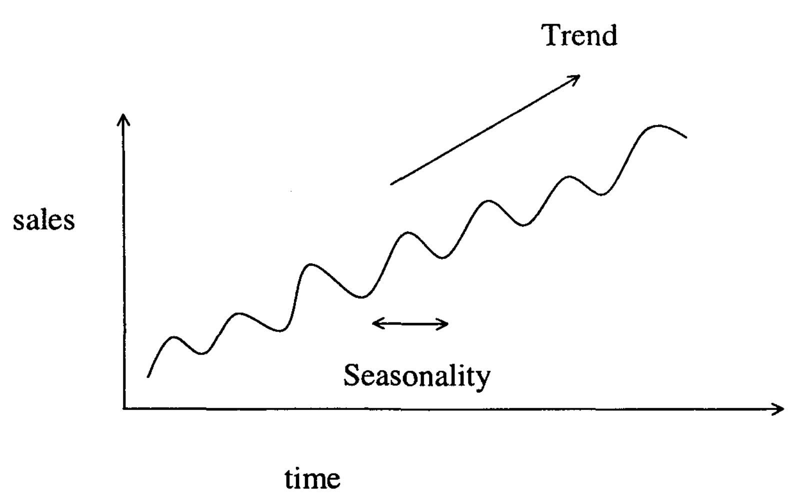
- A time series is a sequence of real numbers, representing the measurements of a real variable at equal time intervals
  - Stock price movements
  - Volume of sales over time
  - Daily temperature readings
  - ECG data
- A time series database is a large collection of time series
  - all NYSE stocks

2

Classical Time Series Analysis
(not the focus of this tutorial)

- Identifying Patterns
  - Trend analysis
    - A company's linear growth in sales over the years
  - Seasonality
    - Winter sales are approximately twice summer sales
- Forecasting
  - What is the expected sales for the next quarter?

3



4

Classical Methods

- Auto-Regressive Moving Average model (ARIMA)
- Exponential Smoothening
- Spectral Decomposition
- etc .....

5

Time Series Problems

(from a databases perspective)

- The Similarity Problem

$X = x_1, x_2, \ldots, x_n$
$Y = y_1, y_2, \ldots, y_n$

Define and compute Sim(X, Y)

E.g. do stocks X and Y have similar movements?

6

- Similarity measure should allow for imprecise matches
- Similarity algorithm should be very efficient
- It should be possible to use the similarity algorithm efficiently in other computations, such as
  - Indexing
  - Subsequence similarity
  - clustering
  - rule discovery
  - etc….

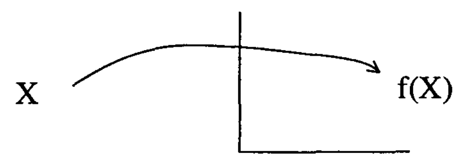
7

- Indexing problem
  - Find all lakes whose water level fluctuations are similar to X
- Subsequence Similarity Problem
  - Find out other days in which stock X had similar movements as today
- Clustering problem
  - Group regions that have similar sales patterns
- Rule Discovery problem
  - Find rules such as "if stock X goes up and Y remains the same, then Z will shortly go down"

8

- Basic approach to the Indexing problem

  Extract a few key "features" for each time series
  Map each time sequence X to a point f(X) in the (relatively low dimensional) "feature space", such that the (dis) similarity between X and Y is approximately equal to the Euclidean distance between the two points f(X) and f(Y)



  Use any well-known spatial access method (SAM) for indexing the feature space

9

- Time series problems are special cases of multimedia indexing problems, such as
  - Retrieve all video clips similar to clip X
- Scalability an important issue
  - If similarity measures, time series models, etc. become more sophisticated, then the other problems (indexing, clustering, etc.) become prohibitive to solve
- Research challenge
  - Design solutions that attempt to strike a balance between accuracy and efficiency

10

Outline of Tutorial

- Part I
  - Discussion of various similarity measures
- Part II
  - Discussion of various solutions to the other problems, such as indexing, subsequence similarity, etc
  - Query language support for time series
  - Miscellaneous issues ...

11

Euclidean Similarity Measure

- View each sequence as a point in n-dimensional Euclidean space (n = length of sequence)
- Define (dis)similarity between sequences X and Y as

$$Lp\ (X, Y)$$

12

Advantages

- Easy to compute

- Allows scalable solutions to the other problems, such as

  • indexing
  • clustering
  • etc...

13

Disadvantages

- Does not allow for different baselines

  • Stock X fluctuates at $100, stock Y at $30

- Does not allow for different scales

  • Stock X fluctuates between $95 and $105, stock Y between $20 and $40

14

Normalization of Sequences

[Goldin and Kanellakis, 1995]

– Normalize the mean and variance for each sequence

Let $\lambda(X)$ and $\rho(X)$ be the mean and variance of sequence X

Replace sequence X by sequence X', where

$X'_i = (X_i - \lambda(X))/\rho(X)$

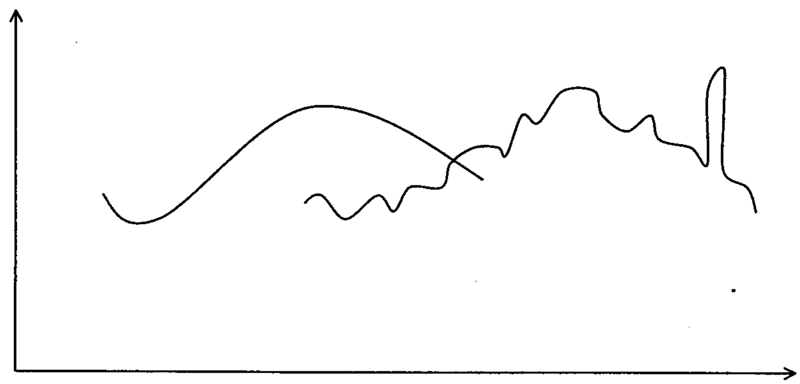
15

Similarity definition still too rigid

- Does not allow for noise or short-term fluctuations
- Does not allow for phase shifts in time
- Does not allow for acceleration-deceleration along the time dimension
- etc ....

16

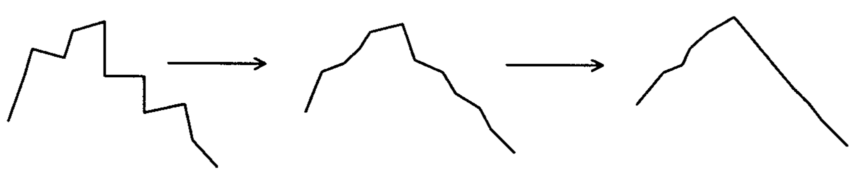
Example



17

A general similarity framework involving a transformation rules language

[Jagadish, Mendelzon, Milo]



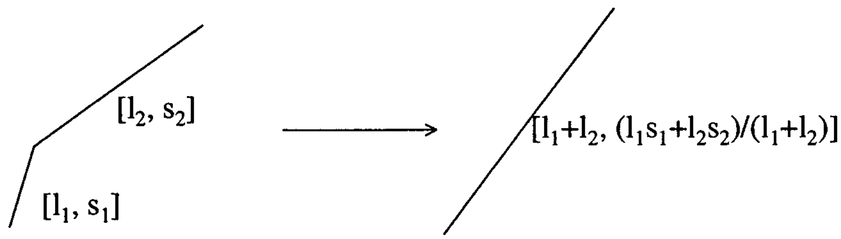
Each rule has an associated cost

18

Examples of Transformation Rules

- Collapse adjacent segments into one segment

new slope = weighted average of previous slopes
new length = sum of previous lengths



19

Combinations of Moving Averages, Scales, and Shifts

[Rafiei and Mendelzon, 1998]

- Moving averages are a well-known technique for smoothening time sequences

  - Example of a 3-day moving average
    $x'_i = (x_{i-1} + x_i + x_{i+1})/3$

20

General Formulation

Let T be a set of transformation rules

Let c(t) be the cost of rule t

$$
\begin{aligned}
D(X, Y) = \min\{\ & D_0(X, Y), \\
& \min_{t \text{ in } T} (c(t) + D(t(X), Y), \\
& \min_{t \text{ in } T} (c(t) + D(X, t(Y)), \\
& \min_{t1,\, t2 \text{ in } T} (c(t_1) + c(t_2) \\
& \qquad + D(t_1(X) + t_2(Y)))\}
\end{aligned}
$$

21

Experiments

[Rafiei & Mendelzon, 1998]

- Stock data from "ftp.ai.mit.edu/pub/stocks/results"
- Euclidean distances combined with moving average transformations produce more intuitive similarity results

22

Disadvantages of Transformation Rules

- Subsequent computations (such as the indexing problem) become more complicated
  - Feature extraction becomes difficult, especially if the rules to apply become dependent on the particular X and Y in question
  - Euclidean distances in the feature space may not be good approximations of the sequence distances in the original space

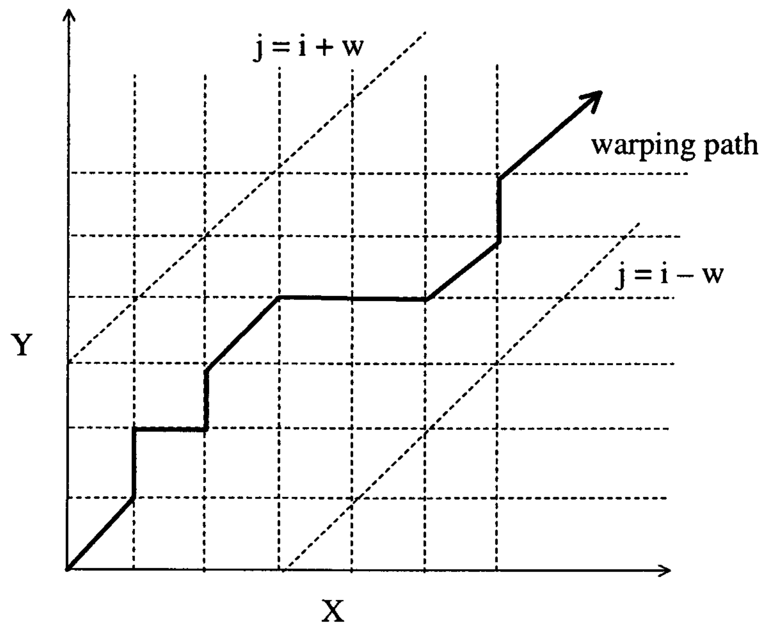
23

Dynamic Time Warping

[Berndt, Clifford, 1994]

- Extensively used in speech recognition
- Allows acceleration-deceleration of signals along the time dimension
- Basic idea
  - Consider $X = x_1, x_2, \ldots, x_n$, and $Y = y_1, y_2, \ldots, y_n$
  - We are allowed to extend each sequence by repeating elements
  - Euclidean distance now calculated between the extended sequences X' and Y'

24



25

Restrictions on Warping Paths

- Monotonicity
  - Path should not go down or to the left
- Continuity
  - No elements may be skipped in a sequence
- Warping Window

  $|i - j| <= w$
- Others ....

26

Formulation

- Let D(i, j) refer to the dynamic time warping distance between the subsequences

$x_1, x_2, \ldots, x_i$

$y_1, y_2, \ldots, y_j$

$$D(i, j) = | x_i - y_j | + \min \{ D(i-1, j), D(i-1, j-1), D(i, j-1) \}$$

27

Solution by Dynamic Programming

- Basic implementation = $O(n^2)$ where n is the length of the sequences
  - will have to solve the problem for each (i, j) pair
- If warping window is specified, then O(nw)
  - Only solve for the (i, j) pairs where $| i - j | <= w$

28

Experiments

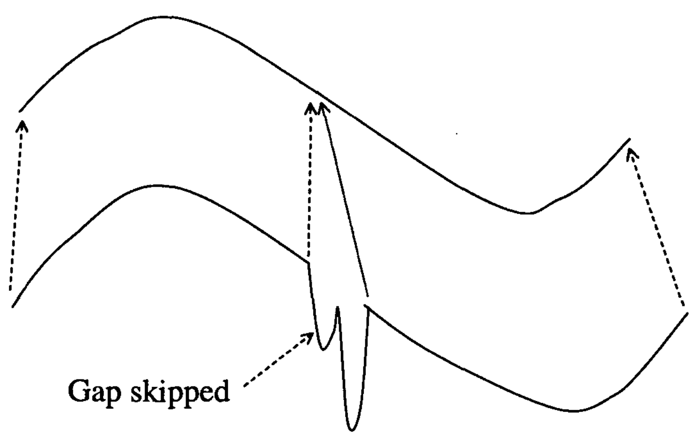
- Predator/Prey Experiments
  - Lynx population fluctuations are related to hare population fluctuations

- Stock market experiments
  - DJIA searched for various types of interesting "template" patterns

29

Longest Common Subsequence Measures

(Allowing for Gaps in Sequences)



30

Basic LCS Idea

X	=	3, **2**, **5**, **7**, 4, 8, **10**, 7
Y	=	**2**, **5**, 4, **7**, 3, **10**, 8, 6
LCS	=	**2, 5, 7, 10**

Sim(X,Y) = |LCS|

Shortcomings

Different scaling factors and baselines (thus need to scale, or transform one sequence to the other)

Should allow tolerance when comparing elements (even after transformation)

31

- Longest Common Subsequences
  - Often used in other domains
    - Speech Recognition
    - Text Pattern Matching
  - Different flavors of the LCS concept
    - Edit Distance
    - Levenshtein Distance

32

LCS-like measures for time series

- Subsequence comparison without scaling [Yazdani & Ozsoyoglu, 1996]
- Subsequence comparison with local scaling and baselines [Agrawal et. al., 1995]
- Subsequence comparision with global scaling and baselines [Das et. al., 1997]
- Global scaling and shifting [Chu and Wong, 1999]

33

LCS without Scaling

[Yazdani & Ozsoyoglu, 1996]

Let Sim(i, j) refer to the similarity between the sequences

$x_1, x_2, \ldots, x_i$ and $y_1, y_2, \ldots. y_j$

Let d be an allowed tolerance, called the "threshold distance"

If $| x_i - y_j | < d$ then

$\mathrm{Sim}(i, j) = 1 + D(i - 1, j - 1)$

else $\mathrm{Sim}(i, j) = \max\{D(i - 1, j), D(i, j - 1)\}$

34

- Scaling
  - Authors do not directly discuss scaling issues
  - A standard global scaling transformation (such as [Goldin et al.'s]) can be used prior to similarity computations
- Complexity of Similarity Algorithm
  - Standard dynamic programming formulation runs in $O(n^2)$ time
  - Improvements possible under certain assumptions on the input

35

LCS-like Similarity with Local Scaling

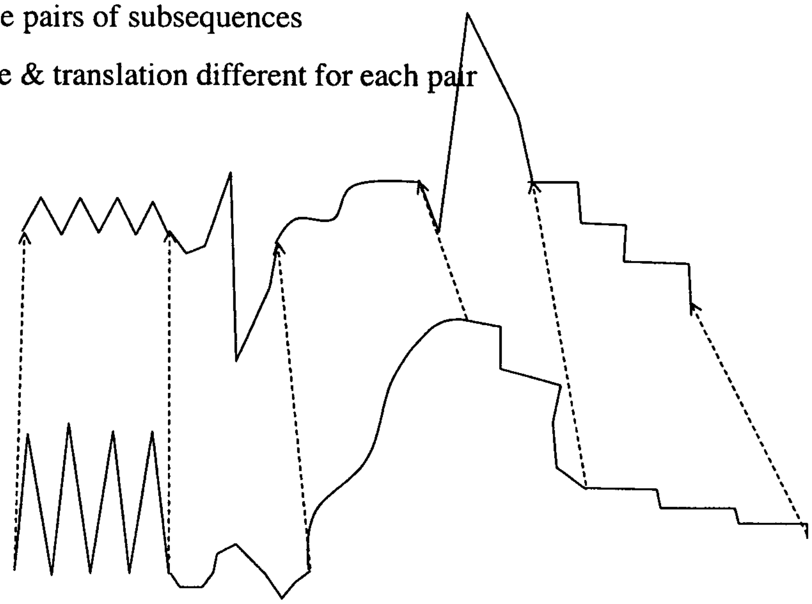
[Agrawal et al, 1995]

- Basic Ideas
  - Two sequences are similar if they have enough non-overlapping time-ordered pairs of subsequences that are similar
  - A pair of subsequences are similar if one can be scaled and translated appropriately to approximately resemble the other

36

Three pairs of subsequences

Scale & translation different for each pair



37

The Algorithm

- Find all pairs of atomic subsequences in X and Y that are similar
  - atomic implies of a certain minimum size (say, a parameter w)
- Stitch similar windows to form pairs of larger similar subsequences
- Find a non-overlapping ordering of subsequence matches having the longest match length

38

Analysis

- Finding all atomic similar subsequence pairs can be done by a spatial self-join (using a SAM such as a R-tree) over the set of all atomic windows
- Window stitching and subsequence ordering problems can be reduced to finding longest paths in a directed acyclic graph
  - Nodes of the DAG are each pair of matching windows
  - Algorithm is essentially a reverse topological sort of the DAG

39

LCS-like Similarity with Global Scaling

[Das, Gunopulos and Mannila, 1997]

- Basic idea: Two sequences X and Y are similar if they have long common subsequence X' and Y' such that

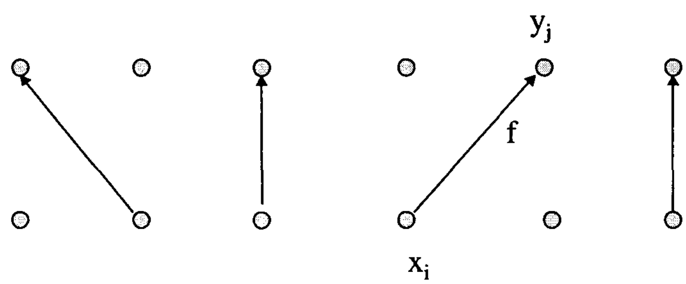
  Y' is approximately = aX' + b

- The scale+translation linear function is derived from the subsequences, and not from the original sequences
  - Thus outliers cannot taint the scale+translation function
- Algorithm
  - Linear-time randomized approximation algorithm

40

Fixed Linear Transformation

Let f : y = ax+b be a given linear transformation



41

Let $Sim_f(X, Y) = |LCS_f| / n$

The overall similarity measure maximizes the above expression over all possible transformations f, thus

$$Sim(X, Y) = \max_{\text{all } f} \{ Sim_f(X, Y) \}$$

42

Computing the Similarity Measure

- Basic LCS computable in $O(n^2)$ time by dynamic programming
- Easily modified to compute Sim_f in $O(n)$ time
  - for a fixed linear transformation f
  - for a fixed "matching window"
- Main task for computing Sim
  - Locate a finite set of all fundamentally different linear transformations
  - Run Sim_f on each f

43

The number of different "unique" linear transformations

$= O(n^2)$

Naïve implementation: run Sim_f on all transformations

Total time taken

$= O(n^3)$

44

Authors show that, of the total $O(n^2)$ possible transformations

a constant fraction of them are "almost as good" as the optimal transformation

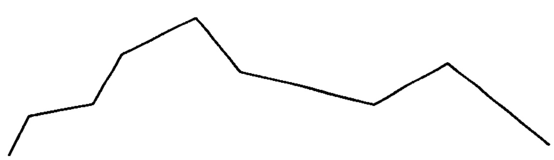
The algorithm just picks a few (constant) number of transformations at random and tries them out

Thus overall running time to get an approximate answer

$= O(n)$

45

Piecewise Linear Representation of Time Series



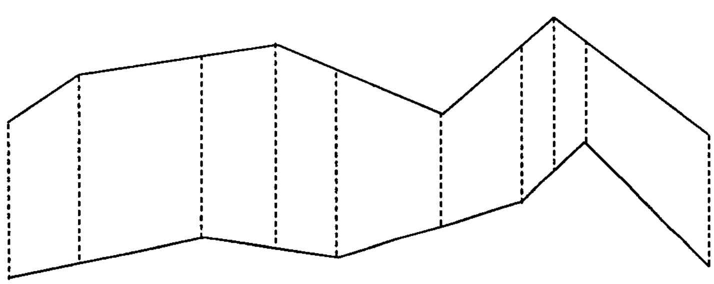
Time series approximated by K linear segments

46

- Such approximation schemes
  - achieve data compression
  - allow scaling along the time axis
- How to select K?
  - Too small => many features lost
  - Too large => redundant information retained
- Given K, how to select the best-fitting segments?
  - Minimize some error function
- These problems pioneered in [Pavlidis & Horowitz 1974], further studied by [Keogh, 1997]

47

Defining Similarity



Distance = (weighted) sum of the difference of projected segments [Keogh & Pazzani, 1998]

48

Probabilistic Approaches to Similarity

[Keogh & Smyth, 1997]

- Probabilistic distance model between time series Q and R
  - Ideal template Q which can be "deformed" (according to a prior distribution) to generate the the observed data R
  - If D is the observed deformation between Q and R, we need to define the generative model
    p(D | Q)

49

- Piecewise linear representation of time series R
- Query Q represented as
  - a sequence of local features (e.g. peaks, troughs, plateaus) which can be deformed according to prior distributions
  - global shape information represented as another prior on the relative location of the local features

50

Properties of the Probabilistic Measure

- Handles scaling and offset translations
- Incorporation of prior knowledge into similarity measure
- Handles noise and uncertainty

51

Probabilistic Generative Modeling Method

[Ge & Smyth, 2000]

- Previous methods primarily "distance based", this method "model based"
- Basic ideas
  - Given sequence Q, construct a model M_Q(i.e. a probability distribution on waveforms)
  - Given a new pattern Q', measure similarity by computing $p(Q'|M_Q)$

52

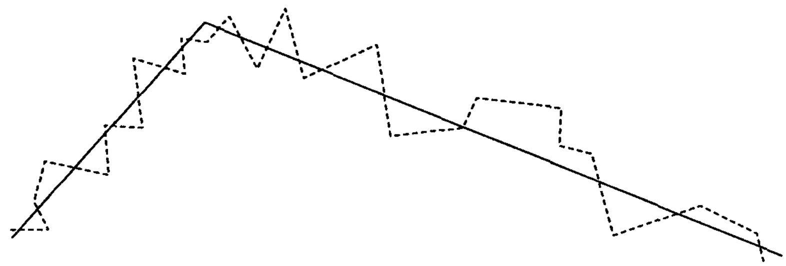
- The model M_Q
  - a discrete-time finite-state Markov model
  - each segment in data corresponds to a state
    - data in each state typically generated by a regression curve
  - a state to state transition matrix is provided

53

- On entering state i, a duration t is drawn from a state-duration distribution p(t)
  - the process remains in state i for time t
  - after this, the process transits to another state according to the state transition matrix

54

Example: output of Markov Model



Solid lines: the two states of the model

Dashed lines: the actual noisy observations

55

Relevance Feedback

[Keogh & Pazzani, 1999]

- Incorporates a user's subjective notion of similarity
- This similarity notion can be continually learned through user interaction
- Basic idea: Learn a user profile on what is different
  - Use the piece-wise linear partitioning time series representation technique
  - Define a Merge operation on time series representations
  - Use relevance feedback to refine the query shape

56

Landmarks

[Perng et. al., 2000]

- Similarity definition much closer to human perception (unlike Euclidean distance)
- A point on the curve is a n-th order landmark if the n-th derivative is 0
  - Thus, local max and mins are first order landmarks
- Landmark distances are tuples (e.g. in time and amplitude) that satisfy the triangle inequality
- Several transformations are defined, such as shifting, amplitude scaling, time warping, etc

57

Part II: Retrieval techniques for time-series

- **The Time series retrieval problem:**
  - Given a set of time series *S*, and a query time series S,
  - find the series that are more similar to S.

- Applications: Time series clustering for:
  - financial, voice, marketing, medicine, video

58

Examples

- Find companies with similar stock prices over a time interval
- Find products with similar sell cycles
- Cluster users with similar credit card utilization
- Cluster products

59

The setting

- Sequence matching or subsequence matching
- Distance metric
- Nearest neighbor queries, range queries, or all-pairs nearest neighbor queries

60

Retrieval algorithms

- We assume that:
  - the similarity function obeys the triangle inequality: $D(A,B) < D(A,C) + D(C,B)$.
  - the query is a full length time series
  - we solve the nearest neighbor query
- We briefly examine the other problems: no distance metric, subsequence matching, all-pairs nearest neighbors

61

Indexing sequences when the triangle inequality holds

- Typical distance metric: L_p norm.
- We use L_2 as an example throughout:
  - $D(S,T) = (\Sigma_{i=1,..,n} (S[i] - T[i])^2)^{1/2}$

62

Simple transformations

- Many distance functions assume that a simple transformation has been applied to the time series [Rafiei, 1999].
- Examples are:
  - Scaling: set minimum = 0, maximum = 1
  - Normalization: set mean = 0, variance = 1

63

Properties of simple transformations

- The transformation is applied once
- It is independent of the distance computation
- The result is another time series

We assume that the transformation has already been performed

64

Indexing time series: The naïve way

- Each time series is an n-dimensional tuple
- Use a high-dimensional index structure to index the tuples
- Such index structures include
  - R-trees,
  - kd-trees,
  - vp-trees,
  - grid-files...

65

High-dimensional index structures

- All require the triangle inequality to hold
- All partition either
  - the space or
  - the dataset into regions
- The objective is to:
  - search only those regions that could potentially contain good matches
  - avoid everything else

66

The naïve approach: Problems

- High-dimensionality:
  - decreases index structure performance (the curse of dimensionality)
  - slows down the distance computation
- Inefficiency

67

Dimensionality reduction

- The main idea: reduce the dimensionality of the space.
- Project the n-dimensional tuples that represent the time series in a k-dimensional space so that:
  - $k << n$
  - distances are preserved as well as possible

68

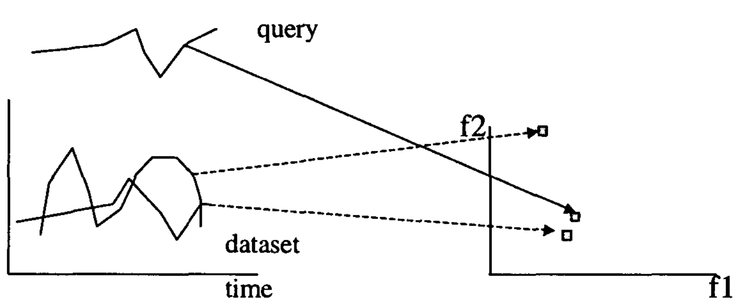
Dimensionality Reduction

- Use an indexing technique on the new space.
- GEMINI ([Faloutsos et al]):
  - Map the query S to the new space
  - Find nearest neighbors to S in the new space
  - Compute the actual distances and keep the closest

69

Dimensionality Reduction

- A time series is represented as a k-dim point
- The query is also transformed to the k-dim space



70

Dimensionality Reduction

- Let F be the dimensionality reduction technique:
  - Optimally we want:
  - D(F(S), F(T)) = D(S,T)
- Clearly not always possible.
- If D(F(S), F(T)) $\neq$ D(S,T)
  - false dismissal (when D(S,T) << D(F(S), F(T)))
  - false positives (when D(S,T) >> D(F(S), F(T)))

71

Dimensionality Reduction

- To guarantee no false dismissals we must be able to prove that:
  - D(F(S),F(T)) < *a* D(S,T)
  - for some constant *a*
- a small rate of false positives is desirable, but not essential

72

What we achieve

- Indexing structures work much better in lower dimensionality spaces
- The distance computations run faster
- The size of the dataset is reduced, improving performance.

73

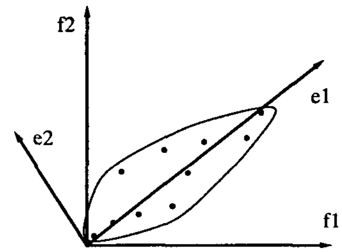
Dimensionality Techniques

- We will review a number of dimensionality techniques that can be applied in this context
  - SVD decomposition,
  - Discrete Fourier transform, and Discrete Cosine transform
  - Wavelets
  - Partitioning in the time domain
  - Random Projections
  - Multidimensional scaling
  - FastMap and its variants

74

SVD decomposition - the Karhunen-Loeve transform

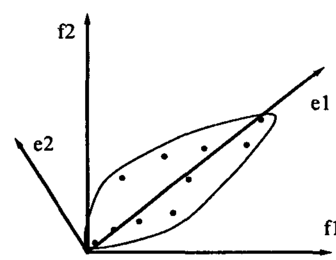
- Intuition: find the axis that shows the greatest variation, and project all points into this axis
- [Faloutsos, 1996]



75

SVD: The mathematical formulation

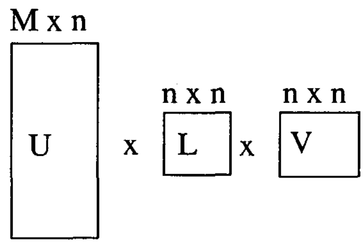
- Find the eigenvectors of the covariance matrix
- These define the new space
- The eigenvalues sort them in "goodness" order



76

SVD: The mathematical formulation, Cont'd

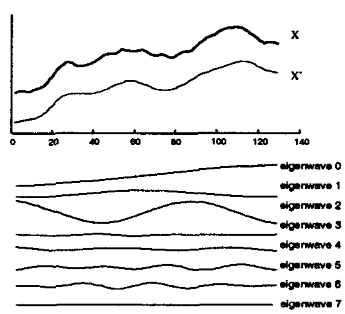
- Let A be the M x n matrix of M time series of length n
- The SVD decomposition of A is: $= U \times L \times V^T$,
  - U, V orthogonal
  - L diagonal
- L contains the eigenvalues of A^TA



77

SVD Cont'd

- To approximate the time series, we use only the k largest eigenvectors of C.
- $A' = U \times L_k$
- A' is an M x k matrix



78

SVD Cont'd

- Advantages:
  - Optimal dimensionality reduction (for linear projections)
- Disadvantages:
  - Computationally hard, especially if the time series are very long.
  - Does not work for subsequence indexing

79

SVD Extensions

- On-line approximation algorithm
  - [Ravi Kanth et al, 1998]
- Local diemensionality reduction:
  - Cluster the time series, solve for each cluster
  - [Chakrabarti and Mehrotra, 2000], [Thomasian et al]

80

Discrete Fourier Transform

- Analyze the frequency spectrum of an one dimensional signal
- For $S = (S_0, \ldots, S_{n-1})$, the DFT is:
- $S_f = 1/\sqrt{n} \sum_{i=0,\ldots,n-1} S_i \, e^{-j2\pi fi/n}$
  $f = 0,1,\ldots n-1,\ j^2 = -1$
- An efficient O(nlogn) algorithm makes DFT a practical method
- [Agrawal et al, 1993], [Rafiei and Mendelzon, 1998]

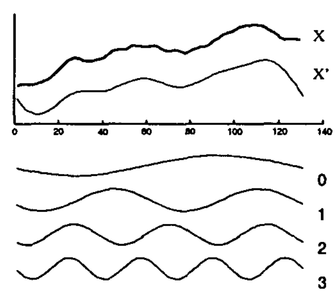
81

Discrete Fourier Transform

- To approximate the time series, keep the k largest Fourier coefficients only.
- Parseval's theorem:

  $\sum_{i=0,\ldots,n-1} S_i^2 = \sum_{i=0,\ldots,n-1} S_f^2$
- DFT is a linear transform so:

  $- \sum_{i=0,\ldots,n-1}(S_i - T_i)^2 = \sum_{i=0,\ldots,n-1}(S_f - T_f)^2$



82

Discrete Fourier Transform

- Keeping k DFT coefficients lower bounds the distance:
  - $\Sigma_{i=0,..,n-1}(S[i]-T[i])^2 > \Sigma_{i=0,..,k-1}(S_f - T_f)^2$
- Which coefficients to keep:
  - The first k (F-index, [Agrawal et al, 1993], [Rafiei and Mendelzon, 1998])
  - Find the optimal set (not dynamic) [R. Kanth et al, 1998]

83

Discrete Fourier Transform

- Advantages:
  - Efficient, concentrates the energy
- Disadvantages:
  - To project the n-dimensional time series into a k-dimensional space, the same k Fourier coefficients must be store for all series
  - This is not optimal for all series
  - To find the k optimal coefficients for M time series, compute the average energy for each coefficient

84

Wavelets

- Represent the time series as a sum of prototype functions like DFT
- Typical base used: Haar wavelets
- Difference from DFT: localization in time
- Can be extended to 2 dimensions
- [Chan and Fu, 1999]
- Has been very useful in graphics, approximation techniques

85

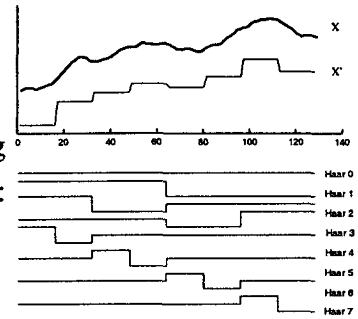
Wavelets

- An example (using the Haar wavelet basis)
  - S ≡ (2, 2, 7, 9) : original time series
  - S' ≡ (5, 6, 0, 2) : wavelet decomp.
  - S[0] = S'[0] - S'[1]/2 - S'[2]/2
  - S[1] = S'[0] - S'[1]/2 + S'[2]/2
  - S[2] = S'[0] + S'[1]/2 - S'[3]/2
  - S[3] = S'[0] + S'[1]/2 + S'[3]/2
- Efficient O(n) algorithm to find the coefficients

86

Using wavelets for approximation

- Keep only k coefficients, approximate the rest with 0
- Keeping the first k coefficients:
  - equivalent to low pass filtering
- Keeping the largest k coefficients:
  - More accurate representation, But not useful for indexing



87

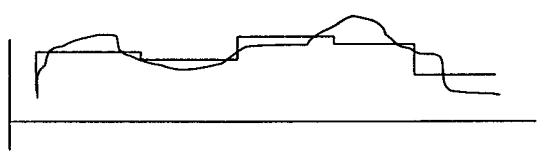
Wavelets

- Advantages:
  - The transformed time series remains in the same (temporal) domain
  - Efficient O(n) algorithm to compute the transformation
- Disadvantages:
  - Same with DFT

88

Line segment approximations

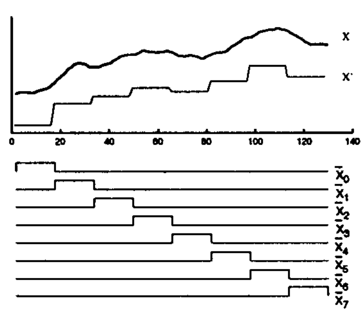
- Piece-wise Aggregate Approximation
  - Partition each time series into k subsequences (the same for all series)
  - Approximate each sequence by :
    - its mean and/or variance: [Keogh and Pazzani, 1999], [Yi and Faloutsos, 2000]
    - a line segment: [Keogh and Pazzani, 1998]



89

Temporal Partitioning

- Very Efficient technique (O(n) time algorithm)
- Can be extended to address the subsequence matching problem
- Equivalent to wavelets (when $k = 2^i$, and mean is used)



90

Random projection

- Based on the Johnson-Lindenstrauss lemma:
- For:
  - $0 < e < 1/2$,
  - any (sufficiently large) set S of M points in R_n
  - $k = O(e^{-2} \ln M)$
- There exists a linear map $f: S \to R_k$, such that
  - $(1-e)\, D(S,T) < D(f(S),f(T)) < (1+e) D(S,T)$ for S,T in S
- Random projection is good with constant probability
- [Indyk, 2000]

91

Random Projection: Application

- Set $k = O(e^{-2} \ln M)$
- Select k random n-dimensional vectors
- Project the time series into the k vectors.
- The resulting k-dimensional space approximately preserves the distances with high probability

- Monte-Carlo algorithm: we do not know if correct

92

Random Projection

- A very useful technique,
- Especially when used in conjunction with another technique (for example SVD)
- Use Random projection to reduce the dimensionality from thousands to hundred, then apply SVD to reduce dimensionality farther

93

Multidimensional Scaling

- Used to discover the underlying structure of a set of items, from the distances between them.
- Finds an embedding in k-dimensional Euclidean that minimizes the difference in distances.
- Has been applied to clustering, visualization, information retrieval...

94

Algorithms for MS

- Input: M time series, their pairwise distances, the desired dimensionality k.
- Optimization criterion:

  stress = $(\Sigma_{ij}(D(S_i,S_j) - D(S_{ki}, S_{kj}))^2 / \Sigma_{ij}D(S_i,S_j)^2)^{1/2}$
  - where $D(S_i,S_j)$ be the distance between time series S_i, S_j, and $D(S_{ki}, S_{kj})$ be the Euclidean distance of the k-dim representations
- Steepest descent algorithm:
  - start with an assignment (time series to k-dim point)
  - minimize stress by moving points

95

Multidimensional Scaling

- Advantages:
  - good dimensionality reduction results (though no guarantees for optimality
- Disadvantages:
  - How to map the query? O(M) obvious solution..
  - slow conversion algorithm

96

FastMap

[Faloutsos and Lin, 1995]

- Maps objects to k-dimensional points so that distances are preserved well
- It is an approximation of Multidimensional Scaling
- Works even when only distances are known
- Is efficient, and allows efficient query transformation

97

How FastMap works

- Find two objects that are far away
- Project all points on the line the two objects define, to get the first coordinate
- Project all objects on a hyperplane perpendicular to the line the two objects define
- Repeat k-1 times

98

MetricMap
[Wang et al, 1999]

- Embeds objects into a k-dim pseudo-metric space
- Takes a random sample of points, and finds the eigenvectors of their covariance matrix
- Uses the larger eigenvalues to define the new k-dimensional space.
- Similar results to FastMap

99

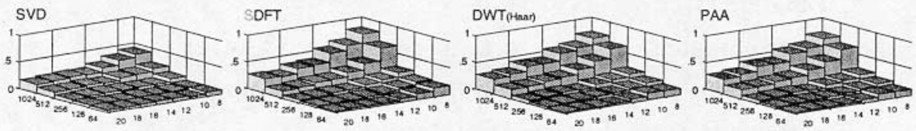
Dimensionality techniques: Summary

- SVD: optimal (for linear projections), slowest
- DFT: efficient, works well in certain domains
- Temporal Partitioning: most efficient, works well
- Random projection: very useful when applied with another technique
- FastMap: particularly useful when only distances are known

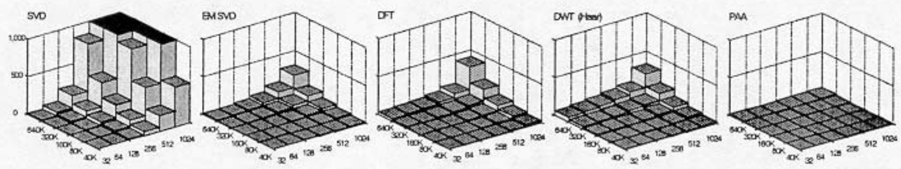
100

An experimental comparison of the techniques [Keogh et al, 2000]

- Accuracy:



- Speed of building the index:



101

Indexing Techniques

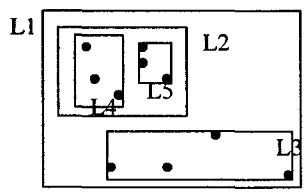
- We will look at:
  - R-trees and variants
  - kd-trees
  - vp-trees and variants
  - sequential scan
- R-trees and kd-trees partition the space, vp-trees and variants partition the dataset, there are also hybrid techniques

102

R-trees and variants

[Guttman, 1984], [Sellis et al, 1987], [Beckmann et al, 1990]

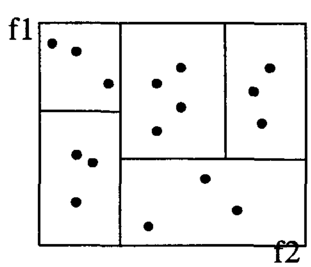
- k-dim extension of B-trees
- Balanced tree
- Intermediate nodes are rectangles that cover lower levels
- Rectangles may be overlapping or not depending on variant (R-trees, R+-trees, R*-trees)
- Can index rectangles as well as points



103

kd-trees

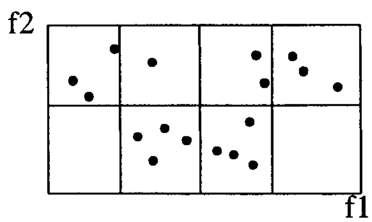
- Based on binary trees
- Different attribute is used for partitioning at different levels
- Efficient for indexing points
- External memory extensions: hB^{Π}-tree



104

Grid Files

- Use a regular grid to partition the space
- Points in each cell go to one disk page
- Can only handle points

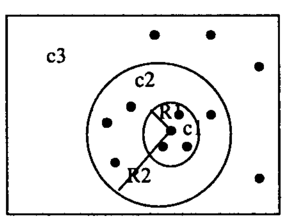


105

vp-trees and pyramid trees

[Ullmann], [Berchtold et al,1998], [Bozkaya et al1997],...

- Basic idea: partition the dataset, rather than the space
- vp-trees: At each level, partition the points based on the distance from a center
- Others: mvp-, TV-, S-, Pyramid-trees



The root level of a vp-tree with 3 children

106

Sequential Scan

- The simplest technique:
  - Scan the dataset once, computing the distances
  - Optimizations: give lower bounds on the distance quickly
  - Competitive when the dimensionality is large.

107

High-dimensional Indexing Methods: Summary

- For low dimensionality (<10), space partitioning techniques work best
- For high dimensionality, sequential scan will probably be competitive with any technique
- In between, dataset partitioning techniques work best

108

The subsequence matching problem

- There is less work on this area
- The problem is more general and difficult
  - [Faloutsos et al, 1994]
  - [Park et al, 2000]
- Most of the previous dimensionality reduction techniques cannot be extended to handle the subsequence matching problem

109

The subsequence matching problem

- If the length of the subsequence is known, two general techniques can be applied:
  - Index all possible subsequences of given length k
    - n-w+1 subsequences of length w for each time series of length n
  - Partition each time series into fewer subsequences, and use an approximate matching retrieval mechanism

110

The bounding rectangle method

- ST-index [Faloutsos et al, 1994]:
  - Find the representation of a sequence window into a feature space (ex. Fourier transform)
  - Use a set of bounding rectangle to bound the trail of a sequence in the feature space
  - Use a Spatial Access Method to index the bounding rectangles
- Typically the trails show smooth movement

111

Similar sequence retrieval when triangle inequality doesn't hold

- In this case indexing techniques do not work (except for sequential scan)
- Most techniques try to speed up the sequential scan by bounding the distance from below.

112

Distance bounding techniques

- Use a dimensionality reduction technique that needs only distances (FastMap, MetricMap, MS)
- Use a pessimistic estimate to bound the actual distance (and accept a number of false dismissals)
- Index the time series dataset using the reduce dimensionality space

113

Example: Time warping and FastMap

[Yi et al, 1998]

- Given M time series
  - Find the M(M-1)/2 distances using the time warping distance measure (does not satisfy the triangle inequality)
  - Use FastMap to project the time series to a k-dim space
- Given a query time series S,
  - Find the closest time series in the FastMap space
  - Retrieve them, and find the actual closest among them
- A heuristic technique: There is no guarantee that false dismissals are avoided

114

The subsequence stitching algorithm

[Agrawal et al, 1995]

- Given a set of M time series with length n,
  - Find all windows of length w that match
    - the triangle inequality holds for windows
    - there O(mn) such windows
  - If two time series have a lot of windows that match, check if these time series are similar.
- An efficient technique to match windows makes the approach work: e-KDB tree [Shim et al, 1997]

115

Shape Queries

- A more general model:
  - Give a general description of a desirable time series
  - Find the time series that conform to this description
- The problems
  - How to define the description (query languages)
  - How to index the time series

116

Query Languages

[Jagadish et al 1995], [Agrawal et al, 1995]

- A general framework to ask similarity based queries:
  - Define a language that describes patterns (ex. regular expressions, or SDL in [Agrawal et al, 1995])
  - Define a set of tranformation rules that can match a sequence to the patterns
  - Define a query language (eg. Relational calculus)
- The framework can be extended to specific applications and distance functions

117

Indexing sequences of images

- When indexing sequences of images, similar ideas apply:
  - If the similarity/distance criterion is a metric,
    Use a dimensionality reduction technique
- [Yadzani and Ozsoyoglu]:
  - Map each image to a set of N features
  - Use a Longest Common Subsequence distance metric to find the distance between feature sequences
  - sim(ImageA, ImageB) = $\Sigma_{i=1..N}$sim(FA_i - FB_i)
- [Lee et al, 2000]:
  - Time warping distance measure
  - Use of Minimum Bounding Rectangles to lower bound the distance

118

Open problems

- Indexing non-metric distance functions
- Similarity models and indexing techniques for higher-dimensional time series
- Efficient trend detection/subsequence matching algorithms

119

Summary

- There is a lot of work in the database community on time series similarity measures and indexing techniques
- Motivation comes mainly from the clustering/unsupervised learning problem
- We look at simple similarity models that allow efficient indexing, and at more realistic similarity models where the indexing problem is not fully solved yet.

120

Bibliography

- Andreas Arning, Rakesh Agrawal, Prabhakar Raghavan: A Linear Method for Deviation Detection in Large Databases. KDD 1996: 164-169.
- R. Agrawal, C. Faloutsos, and A. Swami. Efficient Similarity Search in Sequence Databases. In Int. Conference on Foundations of Data Organization (FODO) 1993.
- R. Agrawal, K.-I. Lin, H.S. Sawhney, K. Shim. Fast Similarity Search in the Presence of Noise, Scaling, and Translation in Time-Series Databases. In Proc. 21th Int. Conf. on Very Large Databases (VLDB-95) 1995.
- R. Agrawal, G. Psaila, E. L. Wimmers, M. Zait: "Querying Shapes of Histories", Proc. of the 21st Int'l Conference on Very Large Databases, Zurich, Switzerland, September 1995.
- Norbert Beckmann, Hans-Peter Kriegel, Ralf Schneider, Bernhard Seeger: The R*-Tree: An Efficient and Robust Access Method for Points and Rectangles. SIGMOD Conference 1990.
- Stefan Berchtold, Christian Böhm, Hans-Peter Kriegel: The Pyramid-Tree: Breaking the Curse of Dimensionality. SIGMOD Conference 1998.
- Donald J. Berndt, James Clifford. Using Dynamic Time Warping to Find Patterns in Time Series. KDD Workshop 1994.
- Donald J. Berndt, James Clifford: Finding Patterns in Time Series: A Dynamic Programming Approach. Advances in Knowledge Discovery and Data Mining 1996: 229-248.
- B. Bollobas, G. Das, D. Gunopulos and H. Mannila Time-Series Similarity Problems and Well-Separated Geometric Sets. In the 13th ACM Symp. on Computational Geometry, 1997, Nice, France.
- Bozkaya, T. and Ozsoyoglu, Z.M., "Indexing Large Metric Spaces for Similarity Search Queries", ACM TODS, 1999.

We would like to thank Eamonn Keogh for allowing us to use some of his figures.

- Bozkaya, T., Yazdani, N. and Ozsoyoglu, Z.M., "Matching and Indexing Sequences of Different Lengths", Proc. Sixth Int. Conf. on Information and Knowledge Management (CIKM), Las Vegas, Nevada, Nov. 1997.
- Bozkaya, T. and Ozsoyoglu, Z.M., "Distance Based Indexing for High Dimensional Metric Spaces", Proc. 1997 ACM SIGMOD International Conference on Management of Data, Tucson, Arizona, May 1997.
- Kaushik Chakrabarti and Sharad Mehrotra, "Local Dimensionality Reduction: A New Approach to Indexing High Dimensional Spaces", VLDB 2000.
- Chan K, Fu W. "Efficient time series matching by wavelets", ICDE 1999.
- K.K.W. Chu, M.H. Wong. Fast Time-Series Searching with Scaling and Shifting. In the 18th ACM Symp. on Principles of Database Systems (PODS-1999)}, Philadelphia, PA
- C.-W. Chung, S.-L. Lee, S.-J. Chun, D.-H. Kim, J.-H. Lee. Similarity Search for Multidimensional Data Sequences, ICDE 2000.
- G. Das, R. Fleisher, L. Gasieniec, D. Gunopulos and J. Karkkainen. Episode Matching. In Combinatorial Pattern Matching 1997, 12-27, Aarhus, Denmark.
- G. Das, D. Gunopulos and H. Mannila. Finding Similar Time Series. In Principles of Data Mining and Knowledge Discovery in Databases (PKDD) 97, Trondheim, Norway.
- Gautam Das, King-Ip Lin, Heikki Mannila, Gopal Renganathan, Padhraic Smyth: Rule Discovery from Time Series. KDD 1998: 16-22.
- C. Faloutsos, M. Ranganathan, and Y. Manolopoulos. Fast Subsequence Matching in Time-Series Databases. In Proc. of 1994 ACM SIGMOD Int. Conf. on Management of Data.
- Christos Faloutsos, King-Ip Lin: FastMap: A Fast Algorithm for Indexing, Data-Mining and Visualization of Traditional and Multimedia Datasets. SIGMOD Conference 1995: 163-174.
- Christos Faloutsos "Searching Multimedia Databases by Content", Kluwer Academic Publishers, 1996.
- Christos Faloutsos, H.V. Jagadish, Alberto O. Mendelzon and Tova Milo A Signature Technique for Similarity-Based Queries SEQUENCES 97, Positano-Salerno, Italy June 11-13 1997.
- A. Guttman. "R-trees: A dynamic index structure for spatial searching", SIGMOD 1984.
- Xianping Ge and Padhraic Smyth "Deformable Markov Model Templates for Time-Series Pattern Matching". SIGKDD 2000.

- Dina Q. Goldin, Paris C. Kanellakis. On Similarity Queries for Time-Series Data: Constraint Specification and Implementation. CP 1995.
- Valery Guralnik, Jaideep Srivastava: Event Detection from Time Series Data. KDD 1999: 33-42.
- S. L. Horowitz,T. Pavlidis: Picture Segmentation by a tree Traversal Algorithm. JACM 23(2): 368-388 (1976).
- Jiawei Han, Guozhu Dong, Yiwen Yin. Efficient Mining of Partial Periodic Patterns in Time Series Database. ICDE 1999: 106-115.
- Imager Wavelet Library. www.cs.ubc.ca/nest/imager/contributions/bobl/wvlt/top.html.
- P. Indyk, M. Gavrilov, R. Motwani, and D. Angelov, Mining Stock Market Data: Cluster Discovery. SIGKDD 2000.
- Piotr Indyk, A. Gionis and R. Motwani"Similarity Search in High Dimensions via Hashing", accepted to the 25th International Conference on Very Large Databases (VLDB) , 1999.
- Piotr Indyk, N. Koudas and S. Muthukrishnan. "Identifying Representative Trends in Massive Time Series Datasets Using Sketches". In the *26th International Conference on Very Large Databases (VLDB)*, 2000.
- P. Indyk and R. Motwani. "Approximate Nearest Neighbors: Towards Removing the Curse of Dimensionality". STOC 98.
- Piotr Indyk "Dimensionality Reduction Techniques for Proximity Problems", accepted to the 11th Symposium on Discrete Algorithms, 2000.
- H.V. Jagadish, A.O. Mendelzon, T. Milo. Similarity-Based Queries. In Symp. on Principles of Database Systems (PODS), 1995.
- H. V. Jagadish, Nick Koudas, S. Muthukrishnan. Mining Deviants in a Time Series Database. VLDB 1999: 102-113.
- W.B. Johnson and J. Lindenstrauss, "Extensions of Lipshitz mapping into Hilbert space", Contemporary Mathematics, 26(1984), 189-206.
- K.V.R. Kanth, D. Agrawal, and A.K. Singh. Dimensionality-Reduction for Similarity Searching in Dynamic Databases. In Proc. of 1998 ACM SIGMOD Int. Conf. on Management of Data, Seattle, WA.
- Kaufman, Rousseauw: "Finding Groups in Dat: an Introduction to Cluster Analysis" John Wiley and Sons, 1995.
- Eamonn J. Keogh, Michael J. Pazzani: An Enhanced Representation of Time Series Which Allows Fast and Accurate Classification, Clustering and Relevance Feedback. KDD 1998: 239-243.
- Eamonn J. Keogh, Michael J. Pazzani: An Indexing Scheme for Fast Similarity Search in Large Time Series Databases. SSDBM 1999

123

- Eamonn J. Keogh, Padhraic Smyth: A Probabilistic Approach to Fast Pattern Matching in Time Series Databases. KDD 1997: 24-30.
- Eamonn Keogh, Kaushik Chakrabarti, Michael Pazzani and Sharad Mehrotra, "Dimensionality Reduction for Fast Similarity Search in Large Time Series Databases", Submitted for publication.
- Keogh, E. (1997). Fast similarity search in the presence of longitudinal scaling in time series databases. In Proceedings of Tools with Artificial. Newport Beach.
- Keogh, E. (1997). A fast and robust method for pattern matching in time series database. In WUSS 1997.
- Keogh & Pazzani, "Scaling up Dynamic Time Warping for Datamining Applications." In 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Boston, 2000.
- Keogh & Pazzani, "A simple dimensionality reduction technique for fast similarity search in large time series databases." In the Fourth Pacific- Asia Conference on Knowledge Discovery and Data Mining. Kyoto, Japan, 2000.
- Keogh & Pazzani, "Scaling up dynamic time warping to massive datasets." In 3rd European Conference on Principles and Practice of Knowledge Discovery in Databases. Prague, 1999.
- Keogh & Pazzani. "Relevance feedback retrieval of time series." The 22th International ACM-SIGIR Conference on Research and Development in Information Retrieval, 1999.
- Flip Korn, H. V. Jagadish, Christos Faloutsos: Efficiently Supporting Ad Hoc Queries in Large Datasets of Time Sequences. SIGMOD Conference 1997: 289-300.
- Vikrant Kobla, David S. Doermann, King-Ip Lin, Christos Faloutsos: Compressed-Domain Video Indexing Techniques Using DCT and Motion Vector Information in MPEG Video. Storage and Retrieval for Image and Video Databases (SPIE) 1997: 200-211.
- King-Ip Lin, H. V. Jagadish, Christos Faloutsos: The TV-Tree: An Index Structure for High-Dimensional Data. VLDB Journal 3(4): 517-542 (1994).
- A. Nanopoulos, Y. Manolopoulos: Indexing Time-Series Databases for Inverse Queries. DEXA 1998.
- S. Park, W. Chu, J. Yoon, C. Hsu. Efficient Similarity Searches for Time-Warped Subsequences in Sequence Databases, ICDE 2000.

124

- C.-S. Perng, H. Wang, S. R. Zhang, D. Stott Parker Landmarks: a New Model for Similarity-based Pattern Querying in Time Series Databases, ICDE 2000.
- D. Rafiei and A. Mendelzon. Similarity-Based Queries for Time-Series Data. In Proc. of 1997 ACM SIGMOD Int. Conf. on Management of Data, 1997, Tuscon, AZ.
- D. Rafiei and Alberto Mendelzon, "Efficient Retrieval of Similar Time Sequences Using DFT" FODO '98 Conference, Kobe, Japan, November 1998.
- Davood Rafiei: On Similarity-Based Queries for Time Series Data. ICDE 1999: 410-417.
- Timos K. Sellis, Nick Roussopoulos, Christos Faloutsos: The R+Tree: A Dynamic Index for Multi-Dimensional Objects. VLDB 1987.
- Dennis Shasha. FinTime --- a financial time series benchmark. http://www.cs.nyu.edu/cs/faculty/shasha/fintime.html
- K. Shim, R. Srikant, R. Agrawal: "High Dimensional Similarity Joins", ICDE 1997 Birmingham, Eng.
- Zbigniew R. Struzik, Arno Siebes: The Haar Wavelet Transform in the Time Series Similarity Paradigm. PKDD 1999.
- A. Thomasian, V. Castello, C.S. Li: Clustering and Singular Value Decomposition for Approximate Indexing in High Dimensional Spaces. CIKM 98.
- J. Wang, X. Wang, K-I Lin, D. Shasha, B. Shapiro, K. Zhang, "Evaluating a class of distance mapping algorithms for Data-Mining and Clustering", KDD 1999.
- White paper: Data Mining Best Practice Papers (Sampling, Fraud Detection). SAS Institute.
- White, Jain "Algorithms and Strategies for Similarity Retrieval" Tech Report VCL-96-101, Visual Computing Laboratory, UC Dais, 1996.
- Christel Wisotzki, Fritz Wysotzki: Prototype, Nearest Neighbor and Hybrid Algorithms for Time Series Classification. ECML 1995: 364-367.
- B. Yi, H.V. Jagadish, C.Faloutsos. Efficient Retrieval of Similar Time Series Under Time Warping. In Int. Conference in Data Engineering, 1998.
- Byoung-Kee Yi, Christos Faloutsos: Fast Time Sequence Indexing for Arbitrary Lp Norms. VLDB 2000.
- Yazdani, N, Bozkaya, T. and Ozsoyoglu, Z.M., "Similarity Search for Sequences of Different Lengths: Matching and Indexing", (submitted for publication), 1998.
- Yazdani, N. and Ozsoyoglu, Z.M., "Sequence Matching of Images", Proc. 8th International Conference on Scientific and Statistical Databases, Stockholm, Sweeden, June 1996.

125