Introduction to Time Series Mining

Slides from Keogh Eamonn’s tutorial:

Your CD-rom contains:
- VLDB 2006 time series tutorial
- More than 100 time series datasets
- Materials for teaching: data mining

“Excellent tutorial concerning temporal mining.” — Ex.: Margaret Crohan, in her book, Data Mining, Introductory and Advanced Topics
What are Time Series?

A time series is a collection of observations made sequentially in time.
Time Series are Ubiquitous! I

People measure things...

- Their blood pressure
- George Bush's popularity rating
- The annual rainfall in Seattle
- The value of their Google stock

...and things change over time...

Thus time series occur in virtually every medical, scientific and businesses domain
Image data, may best be thought of as time series...
Text data, may best be thought of as time series...

The local frequency of words in the Bible

Blue: “God” - English Bible
Red: “Dios” - Spanish Bible

Gray: “El Senor” - Spanish Bible
Video data, may best be thought of as time series...
Why is Working With Time Series so Difficult? Part I

**Answer:** How do we work with very large databases?

- **1 Hour of EKG data:** 1 Gigabyte.
- **Typical Weblog:** 5 Gigabytes per week.
- **Space Shuttle Database:** 200 Gigabytes and growing.
- **Macho Database:** 3 Terabytes, updated with 3 gigabytes a day.

Since most of the data lives on disk (or tape), we need a representation of the data we can efficiently manipulate.
Why is Working With Time Series so Difficult? Part II

**Answer:** We are dealing with subjectivity

The definition of similarity depends on the user, the domain and the task at hand. We need to be able to handle this subjectivity.
Why is working with time series so difficult? Part III

**Answer:** Miscellaneous data handling problems.

- Differing data formats.
- Differing sampling rates.
- Noise, missing values, etc.

We will not focus on these issues in this tutorial.
What do we want to do with the time series data?

- Clustering
- Classification
  - Query by Content
- Motif Discovery
- Rule Discovery
  - 10
  - \( s = 0.5 \)
  - \( c = 0.3 \)
- Visualization
- Novelty Detection
All these problems require **similarity matching**

- Clustering
- Classification
- Motif Discovery
- Rule Discovery
- Query by Content
- Visualization
- Novelty Detection

\[ s = 0.5 \quad c = 0.3 \]
Here is a simple motivation for the first part of the tutorial

You go to the doctor because of chest pains. Your ECG looks strange...

You doctor wants to search a database to find similar ECGs, in the hope that they will offer clues about your condition...

Two questions:

• How do we define similar?

• How do we search quickly?
What is Similarity?
The quality or state of being similar; likeness; resemblance; as, a similarity of features. Webster's Dictionary

Similarity is hard to define, but…
“We know it when we see it”

The real meaning of similarity is a philosophical question.

We will take a more pragmatic approach.
Two Kinds of Similarity

- Similarity at the level of shape
  - Next 40 minutes

- Similarity at the structural level
  - Another 10 minutes

Time series
Euclidean Distance Metric

Given two time series:

\[ Q = q_1 \ldots q_n \]
\[ C = c_1 \ldots c_n \]

\[ D(Q, C) \equiv \sqrt{\sum_{i=1}^{n} (q_i - c_i)^2} \]

About 80% of published work in data mining uses Euclidean distance.
Preprocessing the data before distance calculations

If we naively try to measure the distance between two “raw” time series, we may get very unintuitive results.

Euclidean distance is very sensitive to some “distortions” in the data. For most problems these distortions are not meaningful => should remove them.

In the next few slides we will discuss the 4 most common distortions, and how to remove them:

- Offset Translation
- Amplitude Scaling
- Linear Trend
- Noise
Transformation I: Offset Translation

\[ Q' = Q - \text{mean}(Q) \]
\[ C' = C - \text{mean}(C) \]
\[ D(Q', C') \]
Transformation II: Amplitude Scaling

\[ Q'' = \frac{Q - \text{mean}(Q)}{\text{std}(Q)} \]
\[ C'' = \frac{C - \text{mean}(C)}{\text{std}(C)} \]
\[ D(Q'', C'') \]

Z-score of Q
Removing linear trend:
• fit the best fitting straight line to the time series, then
• subtract that line from the time series.

Removed linear trend
Removed offset translation
Removed amplitude scaling
The intuition behind removing noise is...

Average each datapoint value with its neighbors.

\[ Q' = \text{smooth}(Q) \]
\[ C' = \text{smooth}(C) \]
\[ D(Q', C') \]
A Quick Experiment to Demonstrate the Utility of Preprocessing the Data

Clustered using Euclidean distance on the raw data.

Clustered using Euclidean distance on “clean” data. (removing noise, linear trend, offset translation and amplitude scaling)
The “raw” time series may have distortions which we should remove before clustering, classification etc.

Of course, sometimes the distortions are the most interesting thing about the data, the above is only a general rule.
Dynamic Time Warping

Fixed Time Axis
Sequences are aligned “one to one”.

“Warped” Time Axis
Nonlinear alignments are possible.

Note: We will first see the utility of DTW, then see how it is calculated.
Here is another example on nuclear power plant trace data, to help you develop an intuition for DTW.
Mountain Gorilla
*Gorilla gorilla beringei*

Lowland Gorilla
*Gorilla gorilla graueri*

DTW is needed for most natural objects...
Let us compare Euclidean Distance and DTW on some problems

Leaves

Faces

Sign language

Gun

Trace

Control

2-Patterns

Alexandria

Word Spotting
## Results: Error Rate

Classification using 1-nearest-neighbor
- \( \text{Class}(x) = \text{class of most similar training object} \)

Leaving-one-out evaluation
- For each object: use it as test set, return overall average

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Euclidean</th>
<th>DTW</th>
</tr>
</thead>
<tbody>
<tr>
<td>Word Spotting</td>
<td>4.78</td>
<td>1.10</td>
</tr>
<tr>
<td>Sign language</td>
<td>28.70</td>
<td>25.93</td>
</tr>
<tr>
<td>GUN</td>
<td>5.50</td>
<td>1.00</td>
</tr>
<tr>
<td>Nuclear Trace</td>
<td>11.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Leaves*</td>
<td>33.26</td>
<td>4.07</td>
</tr>
<tr>
<td>(4) Faces</td>
<td>6.25</td>
<td>2.68</td>
</tr>
<tr>
<td>Control Chart*</td>
<td>7.5</td>
<td>0.33</td>
</tr>
<tr>
<td>2-Patterns</td>
<td>1.04</td>
<td>0.00</td>
</tr>
<tr>
<td>Dataset</td>
<td>Euclidean</td>
<td>DTW</td>
</tr>
<tr>
<td>---------------------</td>
<td>-----------</td>
<td>---------</td>
</tr>
<tr>
<td>Word Spotting</td>
<td>40</td>
<td>8,600</td>
</tr>
<tr>
<td>Sign language</td>
<td>10</td>
<td>1,110</td>
</tr>
<tr>
<td>GUN</td>
<td>60</td>
<td>11,820</td>
</tr>
<tr>
<td>Nuclear Trace</td>
<td>210</td>
<td>144,470</td>
</tr>
<tr>
<td>Leaves</td>
<td>150</td>
<td>51,830</td>
</tr>
<tr>
<td>(4) Faces</td>
<td>50</td>
<td>45,080</td>
</tr>
<tr>
<td>Control Chart</td>
<td>110</td>
<td>21,900</td>
</tr>
<tr>
<td>2-Patterns</td>
<td>16,890</td>
<td>545,123</td>
</tr>
</tbody>
</table>

DTW is two to three orders of magnitude slower than Euclidean distance.
How is DTW Calculated? I

We create a matrix the size of $|Q| \times |C|$, then fill it in with the distance between every pair of point in our two time series.
How is DTW Calculated? II

Every possible warping between two time series, is a path though the matrix. We want the best one:

\[
DTW(Q, C) = \min_{w \in PATHS} \sum_{k=1}^{\mid w \mid} w_k
\]

This recursive function gives us the minimum cost path

\[
\gamma(i,j) = d(q_i, c_j) + \min\{ \gamma(i-1, j-1), \gamma(i-1, j), \gamma(i, j-1) \}
\]

Warping path \( w \)
Let us visualize the cumulative matrix on a real world problem. This example shows 2 one-week periods from the power demand time series.

Note that although they both describe 4-day work weeks, the blue sequence had Monday as a holiday, and the red sequence had Wednesday as a holiday.
Let us visualize the cumulative matrix on a real world problem II
What we have seen so far...

- Dynamic Time Warping gives much better results than Euclidean distance on virtually all problems.

- Dynamic Time Warping is very very slow to calculate!

Is there anything we can do to speed up similarity search under DTW?
Simple Idea: Approximate the time series with some compressed or downsampled representation, and do DTW on the new representation. How well does this work...
... there is strong visual evidence to suggest it works well

There is good experimental evidence for the utility of the approach on clustering, classification, etc
Global Constraints

- Slightly speed up the calculations
- Prevent pathological warpings
Accuracy vs. Width of Warping Window

Warping width that achieves max Accuracy:
- FACE: 2%
- GUNX: 3%
- LEAF: 8%
- Control Chart: 4%
- TRACE: 3%
- 2-Patterns: 3%
- WordSpotting: 3%

W: Warping Width
A global constraint constrains the indices of the warping path \( w_k = (i,j)_k \) such that \( j-r \leq i \leq j+r \)

Where \( r \) is a term defining allowed range of warping for a given point in a sequence.
Two Kinds of Similarity

We are done with shape similarity

Let us consider similarity at the structural level for the next 10 minutes
For long time series, shape based similarity will give very poor results. We need to measure similarly based on high level structure.
The basic idea is to extract global features from the time series, create a feature vector, and use these feature vectors to measure similarity and/or classify.

But which
• features?
• distance measure/learning algorithm?

### Table

<table>
<thead>
<tr>
<th>Feature</th>
<th>Time Series</th>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max Value</td>
<td></td>
<td>11</td>
<td>12</td>
<td>19</td>
</tr>
<tr>
<td>Autocorrelation</td>
<td></td>
<td>0.2</td>
<td>0.3</td>
<td>0.5</td>
</tr>
<tr>
<td>Zero Crossings</td>
<td></td>
<td>98</td>
<td>82</td>
<td>13</td>
</tr>
<tr>
<td>ARIMA</td>
<td></td>
<td>0.3</td>
<td>0.4</td>
<td>0.1</td>
</tr>
<tr>
<td>…</td>
<td></td>
<td>…</td>
<td>…</td>
<td>…</td>
</tr>
</tbody>
</table>
Feature-based Classification of Time-series Data
Nanopoulos, Alcock, and Manolopoulos

- features?
- distance measure/learning algorithm?

Learning Algorithm
multi-layer perceptron neural network

Makes sense, but when we looked at the same dataset, we found we could be better classification accuracy with Euclidean distance!

<table>
<thead>
<tr>
<th>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>mean</td>
</tr>
<tr>
<td>variance</td>
</tr>
<tr>
<td>skewness</td>
</tr>
<tr>
<td>kurtosis</td>
</tr>
<tr>
<td>mean (1&lt;sup&gt;st&lt;/sup&gt; derivative)</td>
</tr>
<tr>
<td>variance (1&lt;sup&gt;st&lt;/sup&gt; derivative)</td>
</tr>
<tr>
<td>skewness (1&lt;sup&gt;st&lt;/sup&gt; derivative)</td>
</tr>
<tr>
<td>kurtosis (1&lt;sup&gt;st&lt;/sup&gt; derivative)</td>
</tr>
</tbody>
</table>
Learning to Recognize Time Series: Combining ARMA Models with Memory-Based Learning

Deng, Moore and Nechyba

- **features?**
- **distance measure/learning algorithm?**

**Distance Measure**

Euclidean distance (between coefficients)

- Use to detect drunk drivers!
- Independently rediscovered and generalized by Kalpakis et. al. and expanded by Xiong and Yeung

**Features**

- The parameters of the Box Jenkins model.

  More concretely, the coefficients of the ARMA model.

  “Time series must be invertible and stationary”
Deformable Markov Model Templates for Time Series Pattern Matching
Ge and Smyth

Part 1

**Features**

The parameters of a Markov Model

The time series is first converted to a piecewise linear model

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>0.4</td>
<td>0.5</td>
</tr>
<tr>
<td>0.4</td>
<td>0.2</td>
<td>0.2</td>
</tr>
<tr>
<td>0.5</td>
<td>0.2</td>
<td>0.3</td>
</tr>
</tbody>
</table>

Variations independently developed by Li and Biswas, Ge and Smyth, Lin, Orgun and Williams etc

There tends to be lots of parameters to tune...

**Distance Measure**

“Viterbi-Like” Algorithm

- features?
- distance measure/learning algorithm?
Deformable Markov Model Templates for Time Series Pattern Matching
Ge and Smyth

Part 2

Features

The parameters of a Markov Model

The time series is first converted to a piecewise linear model

On this problem the approach gets 98% classification accuracy*...

But Euclidean distance gets 100%! And has no parameters to tune, and is tens of thousands times faster...
Compression Based Dissimilarity

(In general) Li, Chen, Li, Ma, and Vitányi: (For time series) Keogh, Lonardi and Ratanamahatana

• features?
• distance measure/learning algorithm?

Distance Measure

Co-Compressibility

\[ CDM (x, y) = \frac{C(xy)}{C(x) + C(y)} \]

Features

Whatever structure the compression algorithm finds...

The time series is first converted to the SAX symbolic representation*
Summary of Time Series Similarity

• If you have *short* time series
  • use DTW after searching over the warping window size

• If you have *long* time series,
  • and you know nothing about your data =>
    try compression based dissimilarity
  • if you do know something about your data =>
    extract features
Anomaly (interestingness) detection

We would like to be able to discover surprising (unusual, interesting, anomalous) patterns in time series.

Note that we don’t know in advance in what way the time series might be surprising.

Also note that “surprising” is very context dependent, application dependent, subjective etc.
Simple Approaches I

Limit Checking
Simple Approaches II

Discrepancy Checking
Early statistical detection of anthrax outbreaks by tracking over-the-counter medication sales

Goldenberg, Shmueli, Caruana, and Fienberg

Discrepancy Checking: Example

- normalized sales
- de-noised
- threshold

Actual value
Predicted value

The actual value is greater than the predicted value, but still less than the threshold, so no alarm is sounded.
• Note that this problem has been solved for text strings

  • You take a set of text which has been labeled “normal”, you learn a Markov model for it.

  • Then, any future data that is not modeled well by the Markov model you annotate as surprising.

• Time series can be easily converted to text
  • Discretization of numerical values

We can use Markov models to find surprises in time series…
Training data

Test data (subset)

Markov model

surprise

These were converted to the symbolic representation. I am showing the original data for simplicity.
In the next slide we will zoom in on this subsection, to try to understand why it is surprising.
Normal Time Series

Surprising Time Series

Normal sequence

Actor misses holster

Briefly swings gun at target, but does not aim

Laughing and flailing hand
Anomaly (interestingness) detection

In spite of the nice example in the previous slide, the anomaly detection problem is wide open.

How can we find interesting patterns...

- Without (or with very few) false positives...
- In truly massive datasets...
- In the face of concept drift...
- With human input/feedback...
- With annotated data...
Time Series Motif Discovery
(finding repeated patterns)

Are there any repeated patterns, of about this length — in the above time series?
Time Series Motif Discovery
(finding repeated patterns)
Why Find Motifs?

- Mining **association rules** in time series requires the discovery of motifs. These are referred to as **primitive shapes** and **frequent patterns**.

- Several time series **classification algorithms** work by constructing typical prototypes of each class. These prototypes may be considered motifs.

- Many time series **anomaly/interestingness detection** algorithms essentially consist of modeling normal behavior with a set of typical shapes (which we see as motifs), and detecting future patterns that are dissimilar to all typical shapes.

- In **robotics**, Oates et al., have introduced a method to allow an autonomous agent to generalize from a set of qualitatively different **experiences** gleaned from sensors. We see these “**experiences**” as motifs.

- In **medical data mining**, Caraca-Valente and Lopez-Chavarrias have introduced a method for characterizing a physiotherapy patient’s recovery based of the discovery of **similar patterns**. Once again, we see these “**similar patterns**” as motifs.

- **Animation and video capture**… (Tanaka and Uehara, Zordan and Celly)
**Definition 1.** *Match:* Given a positive real number $R$ (called *range*) and a time series $T$ containing a subsequence $C$ beginning at position $p$ and a subsequence $M$ beginning at $q$, if $D(C, M) \leq R$, then $M$ is called a *matching* subsequence of $C$.

**Definition 2.** *Trivial Match:* Given a time series $T$, containing a subsequence $C$ beginning at position $p$ and a matching subsequence $M$ beginning at $q$, we say that $M$ is a *trivial match* to $C$ if either $p = q$ or there does not exist a subsequence $M'$ beginning at $q'$ such that $D(C, M') > R$, and either $q < q' < p$ or $p < q' < q$.

**Definition 3.** *K-Motif($n,R$):* Given a time series $T$, a subsequence length $n$ and a range $R$, the most significant motif in $T$ (hereafter called the *$l$-Motif($n,R$)) is the subsequence $C_1$ that has highest count of non-trivial matches (ties are broken by choosing the motif whose matches have the lower variance). The $K^{th}$ most significant motif in $T$ (hereafter called the *$K$-Motif($n,R$)*) is the subsequence $C_K$ that has the highest count of non-trivial matches, and satisfies $D(C_K, C_i) > 2R$, for all $1 \leq i < K$. 

![Diagram](image_url)
OK, we can define motifs, but how do we find them?

The obvious brute force search algorithm is just too slow…

The most reference algorithm is based on a *hot* idea from bioinformatics, *random projection* and the fact that SAX allows use to lower bound discrete representations of time series.

Some Examples of Real Motifs

Motor 1 (DC Current)

Astrophysics (Photon Count)
Motifs Discovery Challenges

How can we find motifs…

- Without having to specify the length/other parameters
- In massive datasets
- While ignoring “background” motifs (ECG example)
- Under time warping, or uniform scaling
- While assessing their significance

Finding these 3 motifs requires about 6,250,000 calls to the Euclidean distance function
There are two kinds of time series prediction

- **Black Box**: Predict tomorrow's electricity demand, given *only* the last ten years electricity demand.

- **White Box (side information)**: Predict tomorrow's electricity demand, given the last ten years electricity demand *and* the weather report, *and* the fact that the world cup final is on and...
Black Box Time Series Prediction

- A paper in SIGMOD 04 claims to be able to get better than 60% accuracy on black box prediction of financial data (random guessing should give about 50%). The authors agreed to test blind on a dataset which I gave them, they again got more than 60%. But I gave them quantum-mechanical random walk data!

- A paper in SIGKDD in 1998 did black box prediction using association rules, more than twelve papers extended the work… but then it was proved that the approach *could* not work*!

Nothing I have seen suggests to me that any non-trivial contributions have been made to this problem. (To be fair, it is a very hard problem)