Time Series - Shapelet/Motif Discovery
Motif
Time Series Motif Discovery

- Finding repeated patterns, i.e., pattern mining.
- Are there any repeated patterns, of length $m$ in the TS?
Why Find Motifs?

- Mining **association rules** in TS requires the discovery of motifs. These are referred to as primitive shapes and frequent patterns.
- Several **TS classifiers** work by constructing typical prototypes of each class. These prototypes may be considered motifs.
- Many **TS anomaly detection** algorithms 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.
How do we find Motifs?

• Given a predefined motif length $m$, a brute-force method searches for motifs from all possible comparisons of subsequences.

• It is obviously very slow and computationally expensive.

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

Motif in Bioinformatics

- che-2
- daf-19
- osm-1
- osm-6
- F02D8.3

Motifs:
- GTTGT
- GTT
- GT
- CATGG
- T
- TCCATGG
- AAC
- G
- C
- A
- ACCAT
- AGTAAC
- ACATG
- GTT
- T
- CCAT
- GTT
- GT
- CAT
- G
- GT
The Motif Discovery Algorithm

• **The general problem:**
  • Find the motif $M$ by using a set of sequences called $(w,d)$-motif: sequence of length $w$ that differ from a $d$ points

• **Guiding principle:**
  • Some instances of a motif agree on a subset of positions.

• **Use information from multiple motif instances to construct model**
**k-Projections**

- Choose $k$ positions in string of length $l$.
- Concatenate elements at chosen $k$ positions to form $k$-tuple.
- In $l$-dimensional Hamming space, projection onto $k$ dimensional subspace.

$l = 15$

ATGGCATTCAAGATTC → TGCTGAT

$k = 7$

P = (2, 4, 5, 7, 11, 12, 13)
Random Projection Algorithm

• Choose a projection by selecting $k$ positions uniformly at random.

• For each $l$-tuple in input sequences, hash into bucket based on letters at $k$ selected positions.

• Recover motif from bucket containing multiple $l$-tuples.
Example

• $l = 7$ (motif size), $k = 4$ (projection size)
• Choose projection $(1, 2, 5, 7)$

Input Sequence

...TAGAC\textcolor{red}{ATCCGAC}TTGC\textcolor{red}{CTTAC}TAC...

Buckets

- ATGC
- GCTC
Hashing and Buckets

• Hash function $h(x)$ obtained from $k$ positions of projection.
• Buckets are labeled by values of $h(x)$.
• *Enriched buckets*: contain at least $s$ $l$-tuples, for some parameter $s$. 

ATGC  
GCTC  
CATC  
ATTC
Example of the Motif Discovery Algorithm

• Assume that we have a time series $T$ of length 1,000, and a motif of length 16, which occurs twice, at time $T_1$ and time $T_{58}$.
Example of the Motif Discovery Algorithm

- A mask \{1,2\} was randomly chosen, so the values in columns \{1,2\} were used to project matrix into buckets.
- Collisions are recorded by incrementing the appropriate location in the collision matrix.
Example of the Motif Discovery Algorithm

• A mask \{2,4\} was randomly chosen, so the values in columns \{2,4\} were used to project matrix into buckets.

• Once again, collisions are recorded by incrementing the appropriate location in the collision matrix.
Motif Refinement

- How do we recover the motif from the sequences in the enriched buckets?
- $k$ symbols are known from hash value of bucket.
- Use information in other $l-k$ positions as starting point for local refinement scheme, e.g. EM

ATCCGAC
ATGAGGC
ATAAGTC
ATGTGAC

Local refinement algorithm

ATGCGTC
Candidate motif
Frequency Matrix Model from Bucket

```
| ATCCGAC | ATGAGGC | ATAAAGTC | ATGTGAC |
+---------+---------+----------+---------|
```

Frequency matrix $W$

$$
\begin{align*}
A & = \begin{pmatrix}
1 & 0 & 0.25 & 0.5 & 0 & 0.5 & 0 \\
C & = \begin{pmatrix}
0 & 0 & 0.25 & 0.25 & 0 & 0 & 1 \\
G & = \begin{pmatrix}
0 & 0 & 0.5 & 1 & 0 & 0.25 & 0 \\
T & = \begin{pmatrix}
0 & 1 & 0 & 0.25 & 0 & 0.25 & 0
\end{pmatrix}
\end{pmatrix}
\end{pmatrix}
\end{align*}
$$

EM algorithm

Refined matrix $W^*$
EM Motif Refinement

• For each bucket $h$ containing more than $s$ sequences, form weight matrix $W_h$

• Use EM algorithm with starting point $W_h$ to obtain refined weight matrix model $W_h^*$

• For each input sequence $x(i)$, return $l$ tuple $y(i)$ which maximizes likelihood ratio:
  \[ \frac{Pr(y(i) \mid W_h^*)}{Pr(y(i) \mid P_0)} \]

• $T = \{y(1), y(2), ..., y(N)\}$

• $C(T)$ = consensus string
Expectation Maximization (EM)

- \( S = \{ x(1), \ldots, x(N) \} \): set of input sequences
- **Given:**
  - \( W \): An initial probabilistic motif model
  - \( P_0 \): background probability distribution.
- Find value \( W_{\text{max}} \) that maximizes likelihood ratio:

\[
\frac{\Pr(S \mid W_{\text{max}}, P_0)}{\Pr(S \mid P_0)}
\]

- EM is local optimization scheme. Requires starting value \( W \)
A Single Iteration

• Choose a random $k$-projection.

• Hash each $l$-subsequences $x$ in input sequence into bucket labelled by $h(x)$.

• From each bucket $B$ with at least $s$ sequences, form weight matrix model, and perform EM refinement.

• Candidate motif is the best one found from refinement of all enriched buckets.
Matrix Profile

• The Matrix Profile (MP) is a data structure that annotates a TS and can be exploited for many purposes: e.g. efficient Motif Discovery.
• Given a time series, $T$ and a desired subsequence length, $m$. 
We can use sliding window of length $m$ to extract all subsequences of length $m$. 

$|T| - m + 1$
We can then compute the pairwise distance among these subsequences.

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>7.6952</th>
<th>7.7399</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.6952</td>
<td>0</td>
<td>7.7106</td>
<td>...</td>
<td></td>
</tr>
<tr>
<td>7.7399</td>
<td>7.7106</td>
<td>0</td>
<td>...</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

...
Matrix Profile

- For each subsequence we keep only the distance with the closest nearest neighbor.

<table>
<thead>
<tr>
<th>Set of all subsequences</th>
<th>Set of corresponding nearest neighbor</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>6.04</td>
</tr>
<tr>
<td></td>
<td>2.88</td>
</tr>
<tr>
<td></td>
<td>0.90</td>
</tr>
<tr>
<td></td>
<td>1.61</td>
</tr>
<tr>
<td></td>
<td>5.69</td>
</tr>
<tr>
<td></td>
<td>1.23</td>
</tr>
<tr>
<td></td>
<td>1.40</td>
</tr>
<tr>
<td></td>
<td>...</td>
</tr>
</tbody>
</table>
The distance to the corresponding nearest neighbor of each subsequence can be stored in a vector called **matrix profile** \( P \).
Matrix Profile

- The index of corresponding nearest neighbor of each subsequence is also stored in a vector called matrix profile index.

The matrix profile value at location $i$ is the distance between $T_i$ and its nearest neighbor.
Matrix Profile

- The MP index allows to find the nearest neighbor to any subsequence in constant time.
- Note that the pointers in the matrix profile index are not necessarily symmetric.
- If A points to B, then B may or may not point to A.
- The classic TS motif: the two smallest values in the MP must have the same value, and their pointers must be mutual.
How to “read” a Matrix Profile

• For relatively low values, you know that the subsequence in the original TS must have (at least one) relatively similar subsequence elsewhere in the data (such regions are “motifs”)

• For relatively high values, you know that the subsequence in the original TS must be unique in its shape (such areas are anomalies).
How to Compute Matrix Profile?

• Given a time series, \( T \) and a desired subsequence length, \( m \).

\[
\begin{array}{cccccccccccccc}
\text{inf} & \text{inf} & \text{inf} & \text{inf} & \text{inf} & \text{inf} & \text{inf} & \text{inf} & \text{inf} & \text{inf} & \text{inf} & \text{inf} & \text{inf} & \text{inf} & \text{inf} & \text{inf} & \text{inf} & \text{inf} \\
\end{array}
\]

Matrix profile is initialized as inf vector

This is just a toy example, so the values and the vector length does not fit the time series shown above
How to Compute Matrix Profile?

• Given a time series, $T$ and a desired subsequence length, $m$.

At the first iteration, a subsequence $T_i$ is randomly selected from $T$. 
How to Compute Matrix Profile?

- Given a time series, $T$ and a desired subsequence length, $m$.

![Graphical representation of a time series $T_i$]

We compute the distances between $T_i$ and every subsequences from $T$ (time complexity $= O(|T| \log(|T|)))$

We then put the distances in a vector based on the position of the subsequences:

```
inf  inf  inf  inf  inf  inf  inf  inf  inf  inf  inf  inf  inf  inf  inf
```

The distance between $T_i$ and $T_1$ (first subsequence) is 3
How to Compute Matrix Profile?

• Given a time series, \( T \) and a desired subsequence length, \( m \).

We compute the distances between \( T_i \) and every subsequences from \( T \) (time complexity = \( O(|T|\log(|T|)) \)). We then put the distances in a vector based on the position of the subsequences.

Let say \( T_i \) happen to be the third subsequences, therefore the third value in the distance vector is 0.
• Given a time series, $T$ and a desired subsequence length, $m$.

Matrix profile is updated by apply elementwise minimum to these two vectors:

\[
\begin{array}{cccccccccccccccc}
\text{inf} & \text{inf} & \text{inf} & \text{inf} & \text{inf} & \text{inf} & \text{inf} & \text{inf} & \text{inf} & \text{inf} & \text{inf} & \text{inf} & \text{inf} & \text{inf} & \text{inf} & \text{inf} \\
\end{array}
\]

\[
\begin{array}{cccccccccccccccc}
3 & 2 & 0 & 5 & 3 & 4 & 5 & 1 & 2 & 9 & 8 & 4 & 2 & 3 & 4 & 8 & 6 & 2 & 1
\end{array}
\]
How to Compute Matrix Profile?

• Given a time series, $T$ and a desired subsequence length, $m$.

$$
\begin{array}{cccccccccccccccc}
3 & \text{inf} & \text{inf} & \text{inf} & \text{inf} & \text{inf} & \text{inf} & \text{inf} & \text{inf} & \text{inf} & \text{inf} & \text{inf} & \text{inf} & \text{inf} & \text{inf} & \text{inf} \\
\end{array}
$$

Matrix profile is updated by apply elementwise minimum to these two vectors

$$
\begin{array}{cccccccccccccccc}
3 & 2 & 0 & 5 & 3 & 4 & 5 & 1 & 2 & 9 & 8 & 4 & 2 & 3 & 4 & 8 & 6 & 2 & 1 \\
\end{array}
$$
How to Compute Matrix Profile?

• Given a time series, \( T \) and a desired subsequence length, \( m \).

After we finish to update matrix profile for the first iteration

\[
\begin{array}{cccccccccccccc}
3 & 2 & \text{inf} & 5 & 3 & 4 & 5 & 1 & 2 & 9 & 8 & 4 & 2 & 3 & 4 & 8 & 6 & 2 & 1 \\
\end{array}
\]
How to Compute Matrix Profile?

• Given a time series, $T$ and a desired subsequence length, $m$.

In the second iteration, we randomly select another subsequence $T_j$ and it happens to be the 12th subsequences
How to Compute Matrix Profile?

• Given a time series, $T$ and a desired subsequence length, $m$.

Once again, we compute the distance between $T_j$ and every subsequences of $T$
How to Compute Matrix Profile?

• Given a time series, $T$ and a desired subsequence length, $m$.

The same elementwise minimum
How to Compute Matrix Profile?

• Given a time series, $T$ and a desired subsequence length, $m$.

\[
\begin{array}{ccccccccc}
2 & 2 & \text{inf} & 5 & 3 & 4 & 5 & 1 & 2 \\
\end{array}
\]

\[
\begin{array}{ccccccccc}
9 & 8 & 4 & 2 & 3 & 4 & 8 & 6 & 2 \\
\end{array}
\]

The same elementwise minimum

\[
\begin{array}{ccccccccc}
2 & 3 & 1 & 4 & 4 & 3 & 6 & 2 & 1 \\
\end{array}
\]

\[
\begin{array}{ccccccccc}
5 & 8 & 0 & 2 & 3 & 5 & 9 & 4 & 2 \\
\end{array}
\]
How to Compute Matrix Profile?

- Given a time series, $T$ and a desired subsequence length, $m$.

\[
\text{min}_{i \leq j \leq n-m+1} \min_T t_i \quad T_j
\]

<table>
<thead>
<tr>
<th>2</th>
<th>2</th>
<th>inf</th>
<th>5</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>1</th>
<th>2</th>
<th>9</th>
<th>8</th>
<th>4</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>8</th>
<th>6</th>
<th>2</th>
<th>1</th>
</tr>
</thead>
</table>

The same elementwise minimum

| 2 | 3 | 1 | 4 | 4 | 3 | 6 | 2 | 1 | 5 | 8 | 0 | 2 | 3 | 5 | 9 | 4 | 2 | 2 |
How to Compute Matrix Profile?

• Given a time series, $T$ and a desired subsequence length, $m$.

\[
T_j
\]

$m$

\[
\begin{array}{cccccccccccccc}
2 & 2 & 1 & 5 & 3 & 4 & 5 & 1 & 2 & 9 & 8 & 4 & 2 & 3 & 4 & 8 & 6 & 2 & 1 \\
\end{array}
\]

2 3 1 4 4 3 6 2 1 5 8 0 2 3 5 9 4 2 2

The same elementwise minimum
How to Compute Matrix Profile?

• Given a time series, $T$ and a desired subsequence length, $m$.

We repeat the two steps (distance computation and update) until we have used every subsequence.
How to Compute Matrix Profile?

• Given a time series, $T$ and a desired subsequence length, $m$.

There are $|T|$ subsequences and the distance computation is $O(|T|\log(|T|))$

The overall time complexity is $O(|T|^2\log(|T|))$
Motif Discovery From Matrix Profile

time series, $T$

matrix profile, $P$

Local minimums are corresponding to motifs
Motif Discovery From Matrix Profile

- It is sometime useful to think of time series subsequences as points in m-dimensional space.
- In this view, dense regions in the m-dimensional space correspond to regions of the time series that have a low corresponding MP.
Top-K Motifs

- We need a parameter $R$.
- $1 < R < \text{(small number, say 3)}$
- Let’s make $R = 2$ for now.
- We begin by finding the nearest pair of points, the *motif pair*....
- This pair of subsequences correspond to lowest pair of values in the MP
Top-K Motifs

- We find the nearest pair of points are D1 apart.
- Let’s draw a circle, D1 times R, around both points.
- Any points that are within either of these circles, are added to this motif, in this case just one.
- The Top-1 motif has three members, it is done.
Now let’s find the Top-2 motif. We find the nearest pair of points, excluding anything from the top motif.

The nearest pair of points are D2 apart.

Let’s draw a circle D2 times R, around both points.

Any points that are within either of these circles, is added to this motif, in this case there are two for a total of four items in the Top-2 Motif.
Top-K Motifs

- We have done with the Top-2 Motif
- Note that we will always have:
  - $D_1 < D_2 < D_3 \ldots D_K$
- **When to stop?** (what is $K$?)
- We could use MDL or a predefined $K$. 
We need a parameter $E$ of subsequences to exclude in the vicinity of the anomaly.

Let’s make $E = 2$ for now.

We begin by finding the subsequence with the highest distance in the MP.

This corresponds to the biggest anomaly.
• Then we look for the $E$ closest subsequences to the anomaly.
• We remove all of them.
• We can use a predefined $K$ or the MDL to stop.
Shapelet
• Given a set $X$ of $n$ time series, $X = \{x_1, x_2, ..., x_n\}$, each time series has $m$ ordered values $x_i = \langle x_{t1}, x_{t2}, ..., x_{tm} \rangle$ and a class value $c_i$.

• The objective is to find a function $f$ that maps from the space of possible time series to the space of possible class values.

• Generally, it is assumed that all the TS have the same length $m$. 
Shapelet-based Classification

1. Represent a TS as a vector of distances with representative subsequences, namely shapelets.

2. Use it as input for machine learning classifiers.
Time Series Shapelets

• Shapelets are TS subsequences which are maximally representative of a class.

• Shapelets can provide interpretable results, which may help domain practitioners better understand their data.

• Shapelets can be significantly more accurate/robust because they are local features, whereas most other state-of-the-art TS classifiers consider global features.
Extract Subsequences of all Possible Lengths
Extract Subsequences of all Possible Lengths
Extract Subsequences of all Possible Lengths
Extract Subsequences of all Possible Lengths
Extract Subsequences of all Possible Lengths
Distance with a Subsequence

- Distance from the TS to the subsequence \( \text{SubsequenceDist}(T, S) \) is a distance function that takes time series \( T \) and subsequence \( S \) as inputs and returns a nonnegative value \( d \), which is the distance from \( T \) to \( S \).

- \( \text{SubsequenceDist}(T, S) = \min(\text{Dist}(S, S')), \text{ for } S' \in S_T^{|S|} \)

- Where \( S_T^{|S|} \) is the set of all possible subsequences of \( T \)

- Intuitively, it is the distance between \( S \) and its best matching location in \( T \).
Testing The Utility of a Candidate Shapelet

• Arrange the TSs in the dataset $D$ based on the distance from the candidate.
• Find the optimal split point that maximizes the information gain (same as for Decision Tree classifiers)
• Pick the candidate achieving best utility as the shapelet
• A TS dataset $D$ consists of two classes, A and B.
• Given that the proportion of objects in class A is $p(A)$ and the proportion of objects in class B is $p(B)$,
• The Entropy of $D$ is: $I(D) = -p(A)\log(p(A)) - p(B)\log(p(B))$.
• Given a strategy that divides the $D$ into two subsets $D_1$ and $D_2$, the information remaining in the dataset after splitting is defined by the weighted average entropy of each subset.
• If the fraction of objects in $D_1$ is $f(D_1)$ and in $D_2$ is $f(D_2)$,
• The total entropy of $D$ after splitting is $\hat{I}(D) = f(D_1)I(D_1) + f(D_2)I(D_2)$. 
Information Gain

• Given a certain split strategy $sp$ which divides $D$ into two subsets $D_1$ and $D_2$, the entropy before and after splitting is $I(D)$ and $\hat{I}(D)$.

• The **information gain** for this splitting rule is:

  $$Gain(sp) = I(D) - \hat{I}(D) = I(D) - f(D_1)I(D_1) + f(D_2)I(D_2).$$

• We use the distance from $T$ to a shapelet $S$ as the splitting rule $sp$. 
Problem

• The total number of candidate is

\[
\sum_{l=\text{MINLEN}}^{\text{MAXLEN}} \sum_{T_i \in D} (|T_i| - l + 1)
\]

• For each candidate you have to compute the distance between this candidate and each training sample (space inefficiency)

• For instance
  • 200 instances with length 275
  • 7,480,200 shapelet candidates
Speedup

- Distance calculations form TSs to shapelet candidates is expensive.
- Reduce the time in two ways
- Distance Early Abandon
  - reduce the distance computation time between two TS
- Admissible Entropy Pruning
  - reduce the number of distance calculations
Distance Early Abandon

• We only need the minimum distance.

• Method
  • Keep the best-so-far distance
  • Abandon the calculation if the current distance is larger than best-so-far.
Admissible Entropy Pruning

- We only need the best shapelet for each class
- For a candidate shapelet
  - We do not need to calculate the distance for each training sample
  - After calculating some training samples, the upper bound of information gain (corresponding to the optimistic scenario) < best candidate shapelet
  - Stop calculation
  - Try next candidate
Motif/Shapelet Summary

• A **motif** is a repeated pattern/subsequence in a given TS.

• A **shapelet** is a pattern/subsequence which is maximally representative of a class with respect to a given dataset of TSs.
References

• Matrix Profile I: All Pairs Similarity Joins for Time Series: A Unifying View that Includes Motifs, Discords and Shapelets. Chin-Chia Michael Yeh et al. 1997
