data Structure project
phase1

May 23, 2017

Ali A. Taheri
Ehsan E. Vakhshoori

Contents

1 time series and time series motifs

time series

Time is the most important factor which ensures success in a business. Its difficult to keep up with the pace of time.
A Time Series is a sequence $T = (t_1, t_2, ..., t_n)$ which is an ordered set of n real valued numbers.
The ordering is typically temporal however other kinds of data such as color distributions shapes and spectrographs also have a well defined ordering and canfruitfully be considered time series for the purpose of indexing and mining.

For example:

- Monthly unemployment rates for the previous five years.
- Daily production at a manufacturing plant for a month.
- Decade-by-decade population of a state of the previous century.

Components of a time series

Trend

The long-term tendency of a series to increase or fall (upward trend or downward trend).

Seasonality

The periodic fluctuation in the time series within a certain period. These fluctuations form a pattern that tends to repeat from one seasonal period to the next one.

Cycles

Long departures from the trend due to factors others than seasonality. Cycles usually occur along a large time interval, and the lengths of time between successive peaks or troughs of a cycle are not necessarily the same.

Irregular movement

The movement left after explaining the trend, seasonal and cyclical movements; random noise or error in a time series.

time series motifs

Time series motifs are pairs of individual time series, or subsequences of a longer time series, which are very similar to each other. Time series motifs are repeated segments in a long time series that, if exist, carry precise information about the underlying source of the time series. Motif discovery in time series data has received significant attention in the data mining community since its inception, principally because, motif discovery is meaningful and more likely to succeed when the data is large.

2 difference between match and trivial match in a time series

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.
The first three definitions are summarized in Figure 2, illustrating a time series of length 500, and two subsequences of length 128.

trivial match

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 $\text{dist}(C, M) > R$, andeither $q < p$ or $p < q$.

3 subsequence motif in a time series

Time series motifs are pairs of individual time series, or subsequences of a longer time series, which are very similar to each other.
The k^{th}-Time Series motif is the k^{th} most similar pair in the database D. The

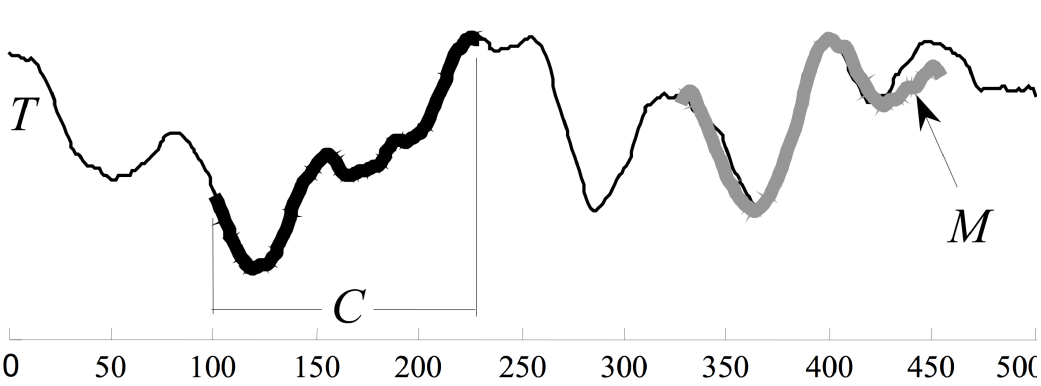

Figure 1: A visual intuition of a time series T (light line), a subsequence C (bold line) and a match M (bold gray line)

pair T_i, T_j is the k^{th} motif iff there exists a set S of pairs of time series of size exactly k-1 such that $\forall t_b$ and $\{t_j, t_d\} \in S$ and $\forall \{t_x, t_y\} \in S$, $\{t_w, t_b\} \notin S$, $dist(t_x, t_y) \leq dist(t_i, t_j) \leq dist(t_a, t_b)$.
Instead of dealing with pairs only we can also extend the notion of motifs to sets of time series that are very similar to each other.
The Subsequence Motif is a pair of subsequences $\{T_{i,n}, T_{j,n}\}$ of a long time series T that are most similar. More formally, $\forall a, b, i, j$ the pair $\{T_{i,n}, T_{j,n}\}$ is the subsequence motif iff $dist(T_{i,n}, T_{j,n}) \leq dist(T_{a,n}, T_{b,n}), |i - j| \geq w$ and $|a - b| \leq w$ for $w > 0$.
Note that we impose a constraint on the relative positions of the subsequence in the motif. This says that there should be a gap of at least w places between the subsequences. This restriction helps to prune out the trivial subsequence motifs. For example (and considering discrete data for simplicity), if we were looking for motifs of length four in the string: sjdbbnvfdfpqoeutyvn-ABABABmbzchslfkeruyousjdq Then in this case we probably dont want to consider the pair {ABAB, ABAB} to be a motif, since they share 50% of their length (i.e AB is common to both). Instead, we would find the pair $\{sjdb, sjdq\}$ to be a more interesting approximately repeated pattern. In this example, we can enforce this by setting the parameter $w = 4$. In all the definitions given above we assumed there is a meaningful way to measure the distance between two time series. There are several such ways in the literature and our method is valid for any distance measure that is a metric. We use Euclidean distance and express it as $d(A, B)$. Recently extensive empirical comparisons have shown that the Euclidean distance is completive with or superior to more complex measures on a wide variety of domains. Furthermore, its competitiveness increases as the datasets get larger, and it

is large datasets that are of interest here.

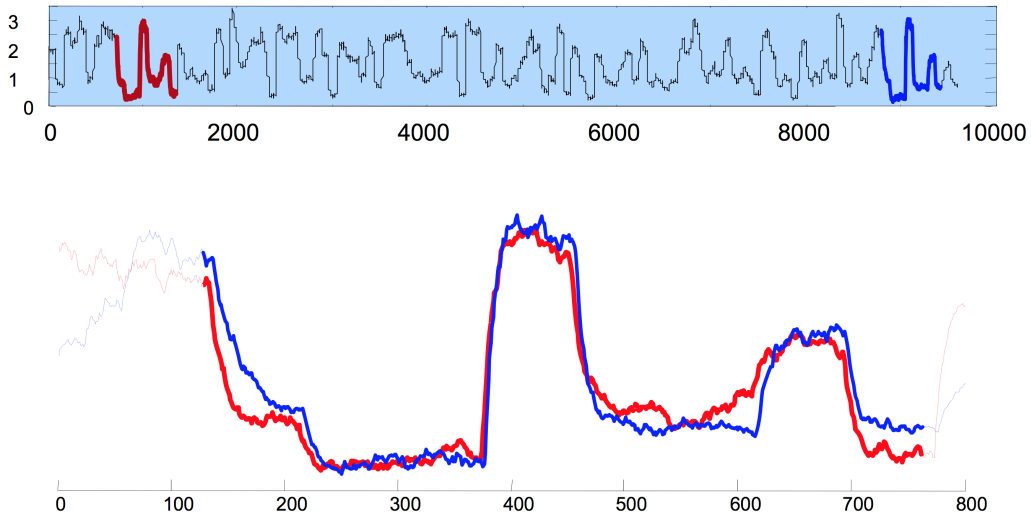

Figure 2: time motives

4 datasets for time series

Generate some input for main algorithm with c++:

(signal):

```
#include <iostream>
#include <ctime>
#include <cstdlib>
#include <fstream>
using namespace std;

int main()
{
  freopen("input1.txt","w",stdout);
  srand(time(NULL));
  for(int i=0; i<1000; i++)
  {
    for(int j=0; j<100; j++)
      cout<<rand()%10<<" ";
    cout<<endl;
  }
  return 0;
}
```

(string):

```cpp
#include <iostream>
#include <ctime>
#include <cstdlib>
#include <fstream>
using namespace std;

int main()
{
  freopen("input2.txt","w",stdout);
  srand(time(NULL));
  for(int i=0; i<1000; i++)
  {
    for(int j=0; j<100; j++)
      cout<<(char)(rand()%26+'a');
    cout<<endl;
  }
  return 0;
}
```

5 algorithm to detect time series motifs

```python
import time

def match(a, b, R):
    count = 0
    for i in range(len(a)):
        if(type(a)is str):
            if(a[i]!=b[i]):
                count = count+1
        else:
            count = count + abs(a[i]-b[i])
    if(count<=R):
        return True
    else:
        return False

def Algorithm_Find_Motif_Brute_Force(T,n,R):
    target = open("output.txt", 'a')
    for i in range (len(T)-n+1):
        count = 0
        pointers = []
        for j in range(i+n,len(T)-n+1):
            if (match(T[i:i+n], T[j:j+n], R)):
                count = count+1
                pointers += [j]
```

```
                target.write(str(T[i:i+n]))
                target.write(" --- ")
                target.write(str(T[j:j+n]))
                target.write("\n")
    target.close()

target = open("output.txt", 'w')
target.close()
tm = open("outputTime.txt",'w')
with open("input2.txt", "r") as ins:
    arr = []
    for line in ins:
        arr.append(line)
for k in [2,5,6]:
    start_time = time.time()
    for i in range(1000):
        string1=arr[i].split()
        Algorithm_Find_Motif_Brute_Force(string1[0], k, 3)
    tm.write(str(time.time()-start_time))
    tm.write("\n")
tm.write("\n")

arr2=[[]]
with open("input1.txt", "r") as ins:
    arr = []
    for line in ins:
        arr.append(line)
for i in range(len(arr)):
    temp = arr[i].split()
    arr2 += [[]]
    for j in range(len(temp)):
        arr2[i] += [ord(temp[j])-48]
for k in[2, 5, 6]:
    start_time = time.time()
    for i in range(len(arr2)):
        Algorithm_Find_Motif_Brute_Force(arr2[i], k ,3 )
    tm.write(str(time.time()-start_time))
    tm.write("\n")
```

6 programs output for a data

Input date: sjdbbnvfdfpqoeutyvnABABABmbzchslfkeruyousjdq

output for this data with K=4 & R=1 :

```
sjdb ---- sjdq
```

output for this data with K=4 & R=2 :

```
sjdb ---- sjdq
euty ---- eruy
vnAB ---- ABAB
nABA ---- BABm
ABAB ---- ABmb
```

output for this data with K=4 & R=3 :

```
sjdb ---- vfdf
sjdb ---- ABmb
sjdb ---- slfk
sjdb ---- sjdq
jdbb ---- fdfp
jdbb ---- ABmb
jdbb ---- Bmbz
dbbn ---- dfpq
dbbn ---- tyvn
dbbn ---- Bmbz
dbbn ---- mbzc
bbnv ---- utyv
bbnv ---- yvnA
bbnv ---- mbzc
bbnv ---- bzch
bnvf ---- tyvn
bnvf ---- vnAB
bnvf ---- bzch
bnvf ---- hslf
nvfd ---- yvnA
nvfd ---- nABA
nvfd ---- slfk
nvfd ---- usjd
vfdf ---- vnAB
vfdf ---- hslf
vfdf ---- lfke
vfdf ---- sjdq
fdfp ---- slfk
fdfp ---- fker
dfpq ---- lfke
dfpq ---- sjdq
fpqo ---- fker
fpqo ---- ruyo
```

```
pqoe —— lfke
pqoe —— uyou
qoeu —— fker
qoeu —— keru
qoeu —— uyou
qoeu —— yous
oeut —— keru
oeut —— eruy
oeut —— yous
oeut —— ousj
euty —— eruy
euty —— ruyo
euty —— ousj
utyv —— ruyo
utyv —— uyou
utyv —— usjd
tyvn —— uyou
yvnA —— BABA
yvnA —— yous
vnAB —— ABAB
nABA —— BABm
ABAB —— ABmb
BABA —— Bmbz
zchs —— yous
chsl —— ousj
hslf —— usjd
slfk —— sjdq
```

7 How much time it takes for algorithm to find motifs?

The brute force algorithm has a time complexity of $O(m^2)$.
This is critical because each distance computation takes $O(k)$ time which makes the brute force algorithm an $O(km^2)$ algorithm