



Advanced Database Systems

1 - D Time Series Data Indexing

Donghyun Jeong



Contents

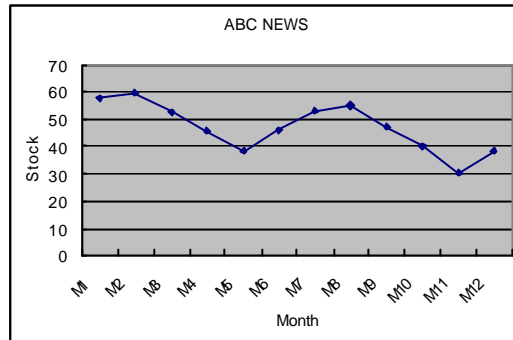
- 1 - D Time Series Data
- Generate Data
- Overall Processing Method
- Preprocessing (I),(II),(III)
- Searching (I), (II)
- Implementation (I), (II)
- Demo & Reference

I'll briefly introduce about 1-D time series data indexing. In fact, I cannot get currently used data such as stock data. Therefore I generate artificial data to process and show how to do indexing. I'll have a look all processes of designed and implemented procedures. (More brief information can find in the book; ADVANCED DATABASE SYSTEMS)



1 -D Time Series Data

- Stock Data(example)



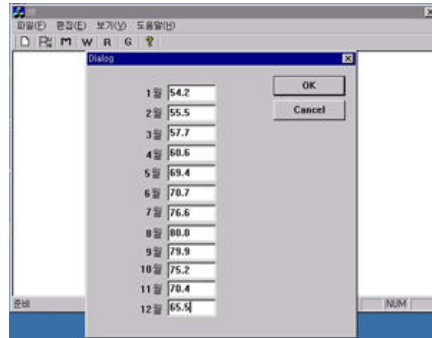
An example of 1-D time series data is stock data. I have designed indexing procedures with computer-generated stock data instead of real data. As you see the histogram on the above, it denotes time series data.

Sample Data (a part)

M1	M2	M3	M4	M5	
	NAME			
57.709375	59.903125	52.584375	45.6375	38.69375 ABC News
69.34375	60.2	61.340625	62.35625	58.6125 MBS TV
69.859375	77.396875	73.921875	67.8875	75.690625 Hallym DB

Generate Data

- 1 - D Time Series Data Generating



As I said on the previous page, time series data(Stock data) generated on special program we made. The generating program made with MFC. Stock data has fluctuations about ± 10 . I generate 100 samples include modeled company names to show it looks like real data. (See on the previous page)



Overall Processing Method

- Preprocessing
 - 1-D Time Series Data -> Spatial Data
 - Using DFT, Clustering(Iterative Method)
 - Indexing (R-tree like method)
- Searching
 - 1-D Time Series Data -> Spatial Data
 - Search nearest data
 - Using Euclidean Distance
 - Refining Process

1-D Time Series Data Indexing has two parts of processing procedures.(Preprocessing/ Searching)

1, Preprocessing

First change 1-D Time Series Data to spatial data on the frequency domain using DFT(Discrete Fourier Transform). Second grouping with similar data using clustering method. Last indexing grouped ID(Index) to get a possibility of searching.

2. Searching

After preprocessing procedure, we can search data which we want. To find data we need to change our data to spatial data using DFT as we used in preprocessing. After get new spatial data which we want to search, calculate Euclidean distance with original spatial data. And then choose a nearest one. In result, we do refining process to get final data.

? see more on next page



Preprocessing (I)

- DFT(Discrete Fourier Transform)

$$c_k = \frac{1}{n} \sum_{h=0}^{n-1} \omega^{-kh} f_h, \quad \omega = e^{2\pi i/n}, k = 1, 2, \dots, n$$

$$e^{\pm i\omega x} = \cos(\omega x) \pm i \sin(\omega x)$$

- No fault dismissals(Parseval's theorem)

$$D_{\text{feature}}(F(x), F(y)) \leq D(x, y)$$

Using DFT, we can change 1-D time series data to spatial data on frequency domain. After change to spatial data, choose three data (f = 1 ~ 3). If the wanted data is included(false alarm), we can find data with the refining process. But if the wanted data is not included(false dismissals), we cannot find data. The reason of changing data to spatial data is that the Euclidean distance of original data is always greater than the Euclidean distance of spatial data which is changed to spatial data using DFT. It denotes no fault dismissals. [2][3]

DFT Pseudo Code

```

for k=0 to sizeofdata -1
  real_tmp = 0 // real data
  imag_tmp = 0 // image data
  for n=0 to sizeofdata -1
    real_tmp = real_tmp + data(n)*cos(2*PI/12*k*n)
    imag_tmp = imag_tmp + -1*data(n)*sin(2*PI/12*k*n)
  next
  power(k) = sqr(real_tmp*real_tmp + imag_tmp*imag_tmp)
  // calculate power of data on frequency domain
next

```



Preprocessing (II)

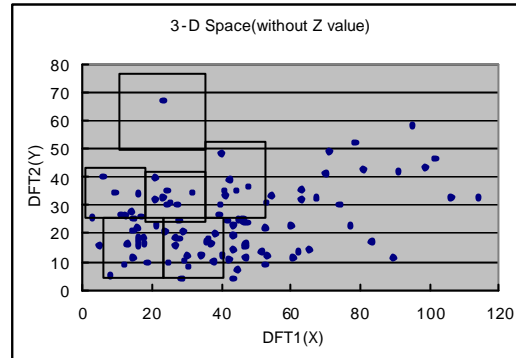
- Clustering
 - Calculate Cluster Center using K-means
 - K-means is a method to search nearest point using Squared Euclidean Distance.
 - Grouping some elements which have a same center value.

I use the iterative method to cluster data. Especially I used K-means algorithm to cluster. K-means algorithm is to make K numbers of cluster center using squared Euclidean distance. [4]

```
----- GROUP LIST(Clustering) -----  
  Cluster Number + {Elements}  
0 { 0 }  
1 { 28 44 87 }  
2 { 19 30 67 84 }  
3 { 3 32 70 }  
4 { 4 39 81 91 95 }  
5 { 5 37 69 83 86 }  
6 { 6 46 48 62 80 93 }  
7 { 7 41 71 74 }  
8 { 8 36 40 63 }  
9 { 9 60 61 78 88 89 90 97 }  
10 { 10 31 54 }  
11 { 11 72 73 92 }  
12 { 12 96 }  
13 { 13 42 49 59 85 }  
14 { 14 35 47 77 }  
15 { 15 26 34 43 55 57 58 65 82 }  
16 { 16 50 64 75 }  
17 { 17 68 94 98 }  
18 { 18 38 51 56 }  
19 { 25 52 66 }
```

Preprocessing (II)

■ Clustering



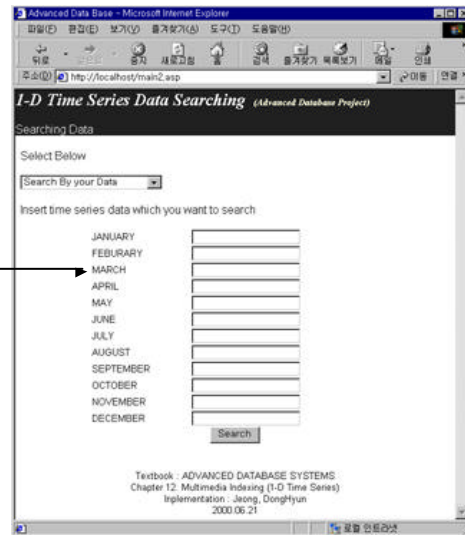
After clustering data as you can see on the above, we can search with cluster center values. The total data(100) changed to 25 cluster center.

----- Cluster Center -----

<u>DFT1</u>	<u>DFT2</u>	<u>DFT3</u>
23.340117	66.733917	10.157041
50.151770	25.601297	4.004073
61.989542	15.195794	22.199620
7.714025	28.789136	24.238079
40.608878	13.905752	7.586689
67.583971	34.087372	15.156011
45.037389	37.401815	18.088304
12.462210	29.154218	6.057019
43.047011	20.739349	31.244777
16.180685	15.162112	11.154085
8.316712	9.745249	3.354249
37.904022	31.389206	11.713221
26.594263	6.602002	19.067617
26.666832	18.493810	13.540259
16.295715	23.074414	19.648849
93.036619	44.201457	28.903012
30.942753	10.509283	10.143203
38.378123	18.404003	17.045466
25.086263	30.154396	19.368506
92.322350	16.700450	20.405816

Searching (I)

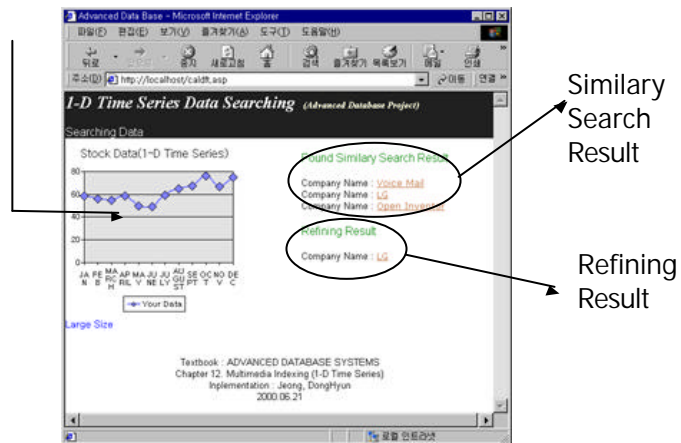
- Search Data



As people input data on the web, the preprocessing procedure change data to spatial data using DFT. With this spatial data, we choose the nearest cluster. It denotes false alarm, we can find a result data after using refining process.

Searching (II)

■ Search Data



As you see on the above, we can find similar search results and refining result. On the refining process, we determined e value must be smaller than 1.0 ($e < 1.0$). Otherwise the exact data which we want to search cannot be found.

Serch

Found Similary Search Result

Company Name : GVE

Company Name : Medal co.

Company Name : YAYAWA

Refining Result

Company Name : Medal co.



Implementation (I)

- Random Generate & insert data
 - Visual C++ 6.0
- User Interface
 - ASP, Msoffice2000 component(Web Application)
- Database
 - SQL Server 7.0

We generate 1-D time series data using random generator in Visual C++. Also we designed user interface using ASP(Active Server Page) to search data on the web. 1-D time series data shows graphical view using graph components in MSOffice200.

In fact we should make database system to test data indexing. Instead of making database system, we use well-known database system(SQLServer 7.0).

Implementation (II)

■ Datagram(table descript)

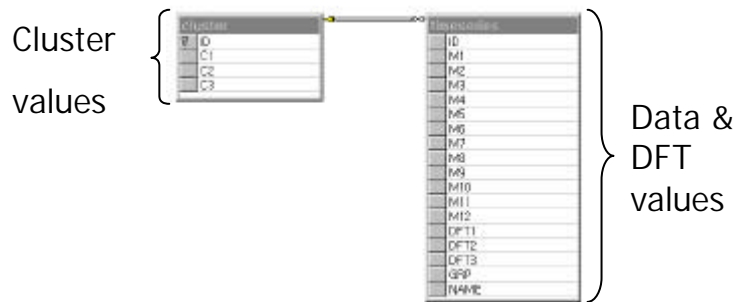


Table consists of cluster data and spatial data. Cluster data and ID are connected with foreign key with GRP(cluster group number).

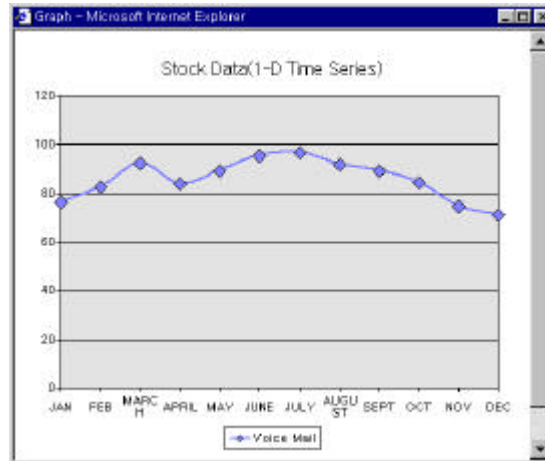
```
CREATE TABLE [dbo].[timeseries] (
    [ID] [float] NULL , [M1] [float] NULL , [M2] [float] NULL , [M3]
[float] NULL , [M4] [float] NULL , [M5] [float] NULL , [M6] [float] NULL , [M7]
[float] NULL , [M8] [float] NULL , [M9] [float] NULL , [M10] [float] NULL , [M11]
[float] NULL , [M12] [float] NULL , [DFT1] [float] NULL , [DFT2] [float] NULL ,
[DFT3] [float] NULL , [GRP] [int] NULL , [NAME] [nvarchar] (255) NULL
) ON [PRIMARY]
GO
```

```
CREATE TABLE [dbo].[cluster] (
    [ID] [int] NOT NULL , [C1] [float] NULL , [C2] [float] NULL , [C3] [float]
NULL ) ON [PRIMARY]
GO
```

```
ALTER TABLE [dbo].[cluster] WITH NOCHECK ADD
    CONSTRAINT [PK_cluster] PRIMARY KEY NONCLUSTERED
    ( [ID] ) ON [PRIMARY]
GO
```

```
ALTER TABLE [dbo].[timeseries] ADD
    CONSTRAINT [FK_timeseries_cluster] FOREIGN KEY
    ( [GRP] ) REFERENCES [dbo].[cluster] ( [ID] )
GO
```

Demo



The image shows web designed 1-D time series data.

MAIN PAGE (overall composition)

- |
- + Search by name
- |
- + Search by data
- | |
- | + DFT & Similar Search & Refining
- |
- + Graph Form Display (graphical display)



Result

- A possibility of 1-D Time Series Data Searching on the Internet.
- Don't know how fast it is compare with another applications.
- Should use real data instead of generated data

We designed and implemented 1-D time series data searching on the web. Actually we has a look a possibility of searching stock data on the web in relevantly short time. In fact there is no 1-D times series data searching product on the web.

The problem is that searching 1-D time series data searching on the web is possible but there is no analysis method to measure the performance of 1-D time series data searching.



Reference

- [1]ADVANCED DATABASE SYSTEMS,Carlo Zaniolo et al. Morgan Kaufmann Publishers, pp.295 -305, 1997.
- [2]Fourier Transform of an image,
<http://www.postech.ac.kr/~yirin/fft/fft.html>
- [3]C++ ALGORITHMS for DIGITAL SIGNAL PROCESSING Second Edition, Paul M. Embree, Damon Danieli, pp.331 - 339, 1998.
- [4]Pattern Recognition with Neural Networks in C++, Abhjit S. Pandya, Robert B. Macy, pp.213-230, 1995.