Query by Content for Time Series Data in RDBMS

INES F. VEGA-LOPEZ

Roadmap

- **Querying non-text data**
- Time series data
- ECG data
- ECG sequence classification
- Extending RDBMS

Non-text data

- Music
- Speech
- Biosignals

- Images
- Video

Querying non-text data

- By describing content
  - Query by associated text
  - Labels, html, etc.
- By content
  - Similarity search
  - Similarity or distance function is required
  - Provided by a domain expert

Roadmap

- Querying non-text data
- **Time series data**
- ECG data
- ECG sequence classification
- Extending RDBMS

Time series data

- A sequence of pairs (t[i], v[i])
  - A timestamp and a value.
  - Delta t is usually constant.
- Sometimes, the absolute time value is not important.
- Then, the time series is just a sequence of values.

Time series data

- Querying have been well studied for the past 20 years
- Two types of queries
  - Whole sequence match
  - Subsequence match

Similarity Search on Time Series Data

- Whole Sequence Match
  - Given a query pattern q of length n, and a DB, B, of sequences of legth n
  - Find all b ∈ B such that

$$Dist(q,b) \leq \varepsilon$$

Similarity Search on Time Series Data

- Sub Sequence Match
  - Given a query pattern q of length n, and a DB, B, of sequences of arbitrary length (each one longer than q)
  - Find all pairs (b, i), b ∈ B, such that

$$Dist(q, b[i : i + n]) \leq \varepsilon$$

How can we do this efficiently?

- For conventional data, we build and index and use it to prune the search space.
  - A linear order exists among the object in the DB.
- For time series, we do not have a linear ordering.
- We can treat a (sub) sequence as a point in *n*-space.
  - *n* is too large
  - Curse of dimensionality

Searching for (sub) sequences

- Generic Multimedia Indexing: GEMINI
  - Map database Objects into a feature space.
  - Index the transformed objects using a SAM
  - Transform query objects to the feature space
  - Search in this feature space
  - Filter out false positives

Mapping into a Feature Space

- DFT
- DWT

- PAA
- APCA

- SAX
- Etc.

Roadmap

- Querying non-text data
- Time series data
- **ECG data**
- ECG sequence classification
- Extending RDBMS

ECG Data

- We want to do KDD on time series.
  - Let us concentrate on a particular domain.
  - Medicine has a high social impact.
  - ECG data has some very interesting challenges.

- Can we build upon existing models?
  - Can we use try and tested RDBMS'?

~~Issues~~ Challenges with ECG data

- An ECG contains more than one signal
  - Usually 2 or 12 leads
- Different ECG's might have different lengths
  - A few minutes to a couple of days
- Different ECG's might have different sampling ratios
  - 128 Hz to 1 or 2 KHz
- Values' bit-depth might also vary among ECG's
  - 8 to 20 bits per value

What about database systems?

- All these characteristics can be captured by the ER model just fine.
- In turn, this model can be transformed into relation.

An instance of an ECG DB

id_paciente	fecha_nacimiento	genero
251257	1927-03-25	F
275917	1938-04-23	M
306936	1962-07-14	F
291713	1934-04-27	F
312304	1968-10-05	F
308056	1954-03-20	F
317911	1931-06-22	M
277371	1977-12-26	F
285170	1947-02-20	M
278173	1946-04-25	M
270014	1936-09-15	M
301829	1992-10-13	F
311457	1932-03-10	F
294203	1991-12-28	M

id_ecg	id_paciente	longitud	fecha_captura	tipo_estudio
1	251257	8399360	2011-03-28	Holter
2	275917	8893782	2011-03-25	Holter
3	306936	8060758	2011-03-29	Holter
4	291713	8432126	2011-03-23	Holter
5	312304	8421206	2011-03-21	Holter
6	308056	8486742	2011-03-21	Holter
7	317911	8311979	2011-03-21	Holter
8	277371	7984299	2011-03-21	Holter
9	285170	8426667	2011-03-17	Holter
10	278173	8262827	2011-03-28	Holter
11	270014	8377515	2011-03-28	Holter
12	301829	8486742	2011-03-22	Holter
13	311457	8055296	2011-03-21	Holter
14	294203	8262827	2011-03-16	Holter

id_derivacion	id_ecg	signal_a
1	1	R0140754-sig-1.bin
2	1	R0140754-sig-2.bin
3	1	R0140754-sig-3.bin
4	2	R0131968-sig-1.bin
5	2	R0131968-sig-2.bin
6	2	R0131968-sig-3.bin
7	3	R0149361-sig-1.bin
8	3	R0149361-sig-2.bin
9	3	R0149361-sig-3.bin
10	4	R0098061-sig-1.bin
11	4	R0098061-sig-2.bin
12	4	R0098061-sig-3.bin
13	5	R0080099-sig-1.bin
14	5	R0080099-sig-2.bin

What needs to be done?

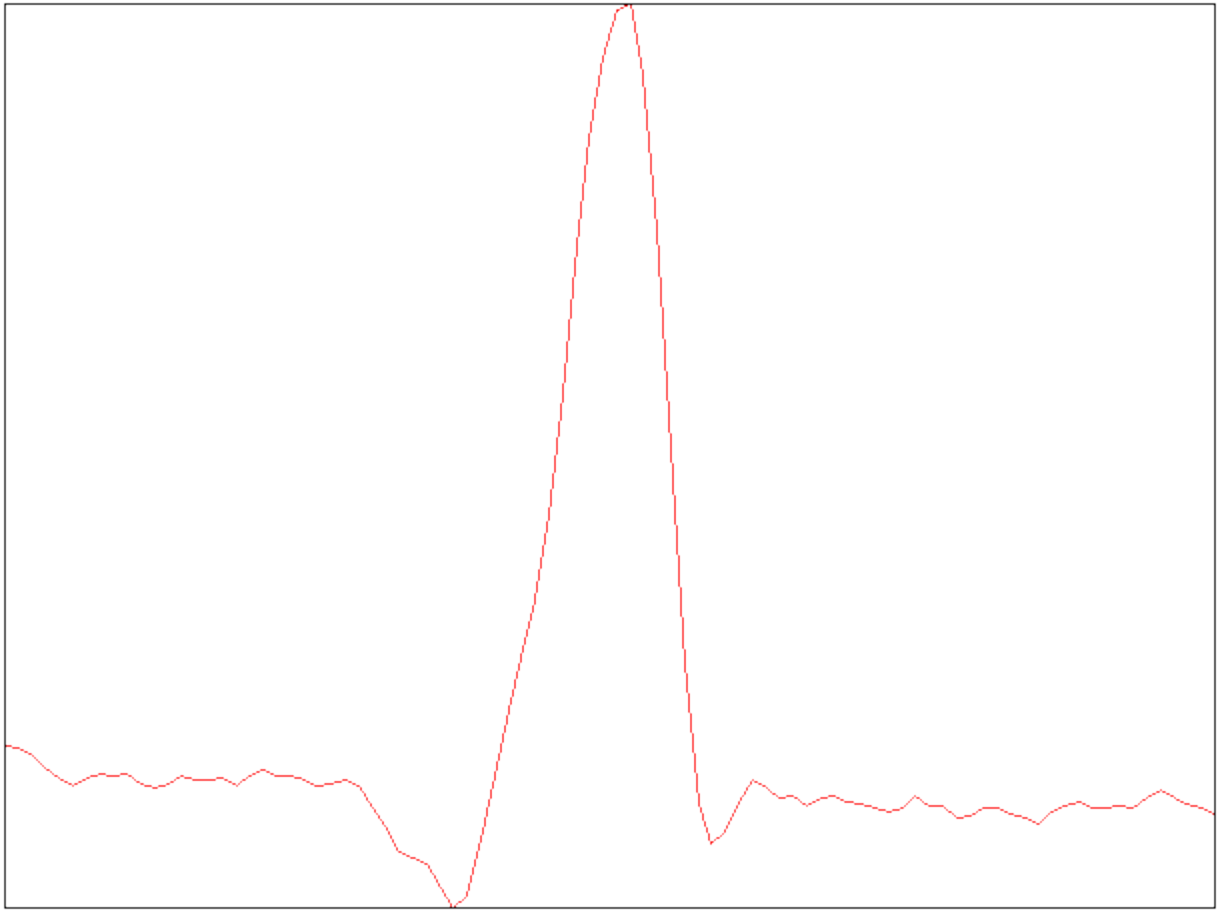
- The content of an ECG signal is not a conventional data type.

- We need to define operators on this type
  - What operators?
    - Similarity Search
    - Define a formal model

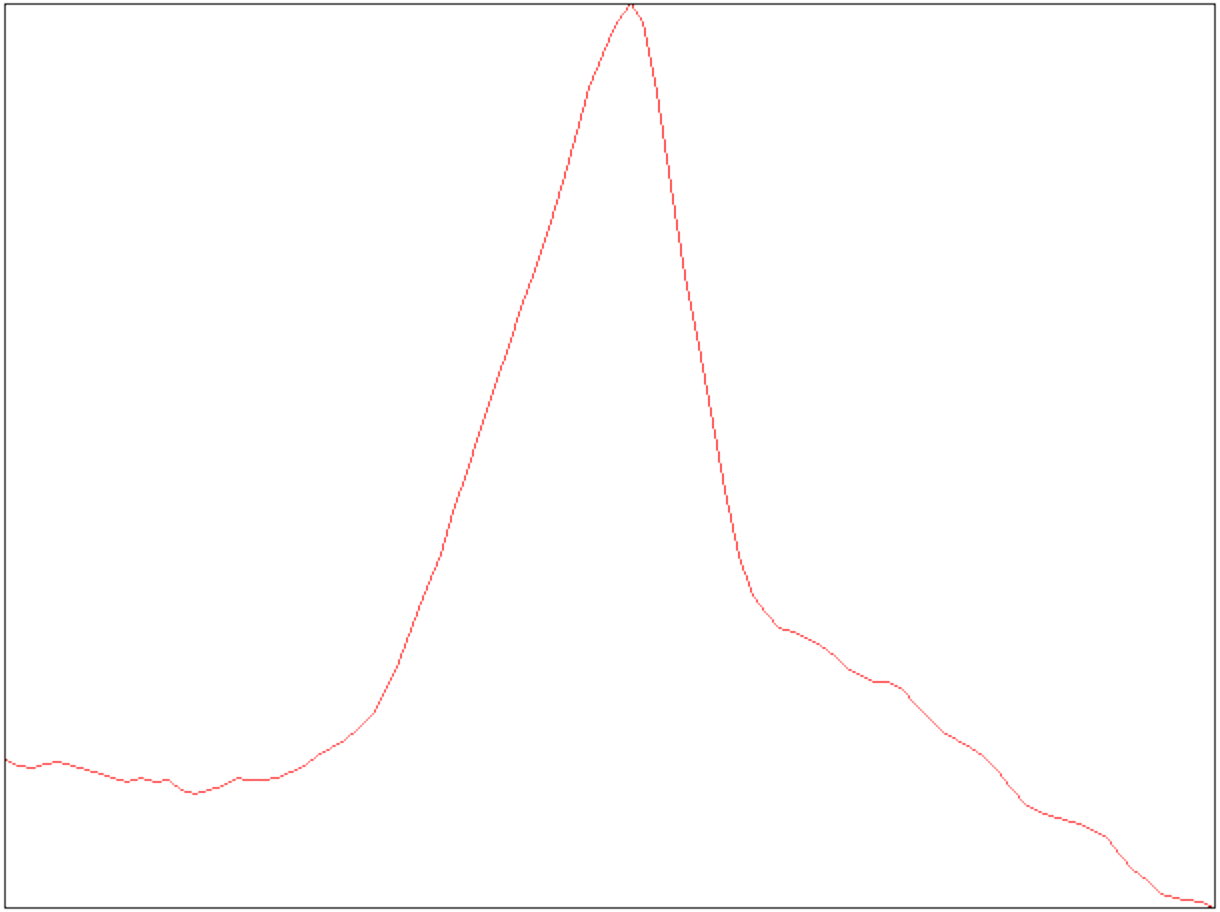
Roadmap

- Querying non-text data
- Time series data
- ECG data
- **ECG sequence classification**
- Extending RDBMS

Normal Hearth Beat



Premature Ventricular Contraction



Similarity Search

- K-nn search
  - This gives us signals and the position of a matching subsequence
- Subsequence retrieval
  - This gives us the content of the matching signal

K-NN Search


```
SELECT NN(D.signal, query_pattern, n)
FROM ECG_DATA D
WHERE <condition>;
```

Sub-sequence Fetch


```
SELECT subsequence(D.signal, position, n)
FROM ECG_DATA D
WHERE D.signal = signal_id;
```

What about the Distance Function?

- For Querying Time Series, the DB community has been using L_p norm.
  - Most often Euclidean
- Cardiologist use Cross Correlation
  - This is not an L_P norm
  - SAM's cannot be used.

Euclidean Distance

$$Dist(X,Y) = \sqrt{\sum_i \left(x[i] - y[i]\right)^2}$$

Cross Correlation Distance

$$Dist(X,Y) = \frac{\sum_i \left|x[i]-\hat{x}\right|\left|y[i]-\hat{y}\right|}{\sqrt{\sum_i \left(x[i]-\hat{x}\right)^2}\sqrt{\sum_i \left(y[i]-\hat{y}\right)^2}}$$

Roadmap

- Querying non-text data
- Time series data
- ECG data
- ECG sequence classification
- **Extending RDBMS**

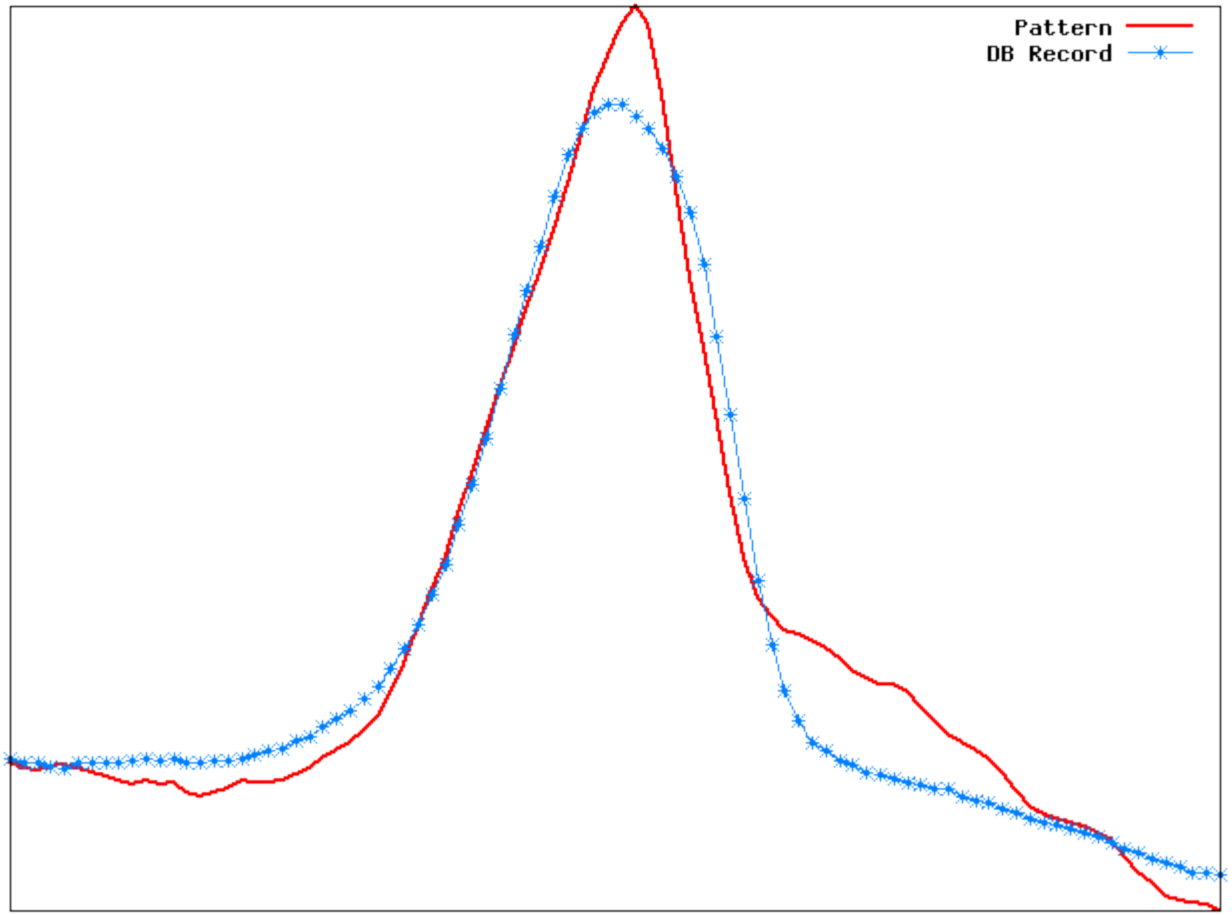
Similarity Searching with UDF


```
SELECT nn_ecg_file(s.valores_archivo, 'latido_ventricular_prematura.bin', 90)
FROM signal_d s;

resultado:
                    nn_ecg_file
---------------------------------------------------
 /fcod-data-mitdb-223-signal-1.bin 561057 2.696500
(1 row)
```

Sub-sequence Fetch


```
SELECT subsequence(s.valores_archivo, 561057, 90)
FROM signal_d s
WHERE s.valores_archivo LIKE '%fcod-data-mitdb-223-signal-1.bin';

resultado:
 subsequence
--------------
 0 -0.580000
 1 -0.595000
 2 -0.595000
 3 -0.605000
 4 -0.610000
 5 -0.595000
 6 -0.595000
 7 -0.595000
 8 -0.590000
 9 -0.585000
 10 -0.575000
 11 -0.585000
 12 -0.580000
 13 -0.595000
 14 -0.595000
 15 -0.585000
--More--(27%)
```

PVC: A Match in the DB



Which distance function is better?

- Using the MIT-BIH Arrhythmia DB
- For healthy – non-healthy classification
  - 98.35 % for Euclidean.
  - 98.59 % For Cross Correlation.
- For pathology classification (15 classes)
  - 97.70 % For Euclidean.
  - 98.14 % For Cross Correlation.
- Too close to call

Are UDF's Efficient?

- We stored ECG signals as BLOBs and as reference to a file.
- We developed an ad-hoc stand alone search application.
  - This uses a file repository.
- Using BLOBs has significant overhead both in storage (5X) and in total elapsed time (10X).
- UDF's on files are as efficient as ad-hoc queries.

Conclusions

- Similarity Search is complex because all data must be scanned.
  - It can be efficiently implemented to extend a RDBMS.
  - Compared to an ad-hoc query.
- It is worth exploring GEMINI.
  - Now that we now that Euclidean distance can be used.
- Data encoding should be considered.
  - We might not be getting much IO savings

Questions

Thanks!