

COMP8400: Algorithms and Techniques for Data Mining

Mining data streams and time series

Characteristics of data streams

- Data streams
 - Continuous, ordered, changing, fast, huge amount
 - Traditional DBMS—data stored in finite, persistent data sets
- Characteristics
 - Huge volumes of continuous data, possibly infinite
 - Fast changing and requires fast, real-time response
 - Data stream captures nicely our data processing needs of today
 - Random access is expensive—single scan algorithm (*can only have one look at each record!*)
 - Store only the summary of the data seen thus far
 - Most stream data are at pretty low-level or multi-dimensional in nature, needs multi-level (ML) and multi-dimensional (MD) processing

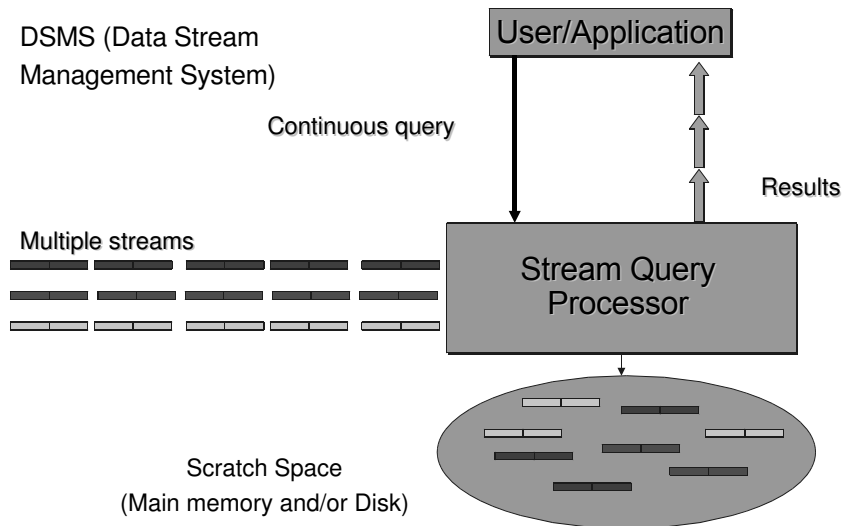
Lecture outline

- Mining data streams
 - Characteristics of data streams
 - Stream data applications
 - Data stream management system
 - Challenges and methodologies of data stream processing
 - Stream data mining versus stream querying
 - Clustering data streams
 - Challenges for mining dynamics in data streams
- Mining time-series data
 - Categories of time-series movements
 - Estimation of trend curve
 - Trend discovery
 - Similarity search in time-series analysis

Stream data applications

- Telecommunication calling records
- Business: credit card transaction flows
- Network monitoring and traffic engineering
- Financial market: stock exchange
- Engineering & industrial processes: power supply and manufacturing
- Sensor, monitoring & surveillance: video streams, RFIDs (Radio Frequency Identification)
- Security monitoring
- Web logs and Web page click streams
- Massive data sets (even saved but random access is too expensive)

Architecture: Stream query processing



Source: Han and Kamber, DM Book, 2nd Ed. (Copyright © 2006 Elsevier Inc.)

DBMS versus DSMS

- Persistent relations
- One-time queries
- Random access
- "Unbounded" disk store
- Only current state matters
- No real-time services
- Relatively low update rate
- Data at any granularity
- Assume precise data
- Access plan determined by query processor, physical DB design
- Transient streams
- Continuous queries
- Sequential access
- Bounded main memory
- Historical data is important
- Real-time requirements
- Possibly multi-GB arrival rate
- Data at fine granularity
- Data stale/imprecise
- Unpredictable/variable data arrival and characteristics

Challenges of stream data processing

- Multiple, continuous, rapid, time-varying, ordered streams
- Main memory computations
- Queries are often continuous
 - Evaluated continuously as stream data arrives
 - Answer updated over time
- Queries are often complex
 - Beyond element-at-a-time processing
 - Beyond stream-at-a-time processing
 - Beyond relational queries (scientific, data mining, OLAP)
- Multi-level/multi-dimensional processing and data mining
 - Most stream data are at low-level or multi-dimensional in nature

Methodologies for stream data processing

- Major challenge
 - Keep track of a large universe (for example, IP address, not ages)
- Methodology
 - Synopses (trade-off between accuracy and storage)
 - Use *synopsis* data structure, much smaller ($O(\log^k N)$ space) than their base data set ($O(N)$ space), with N the number of elements in the stream data
 - Compute an *approximate answer* within a *small error range* (factor ϵ of the actual answer)
- Major methods
 - *Random sampling* (maintain a set of candidates in memory)
 - *Histograms* (approximate frequency distribution of values in stream)
 - *Sliding windows* (make decision based on only recent data)
 - *Multi-resolution models* (balanced trees, wavelets, micro-clusters)
 - *Sketches* (summarises data, can be done in one pass)
 - *Randomised algorithms* (Monte Carlo algorithm, bound on run time)

Stream data mining versus stream querying

- Stream mining is a more challenging task in many cases
 - It shares most of the difficulties with stream querying
 - But often requires less *precision*, for example, no join, grouping, sorting
 - Patterns are hidden and more general than querying
 - It may require exploratory analysis (not necessarily continuous queries)
 - Change in data characteristics: *Concept drift*
- Stream data mining tasks
 - Frequent patterns in data streams (approximate frequent patterns only)
 - Mining outliers and unusual patterns in stream data
 - Classification of stream data (approximate decision trees, classifier ensemble)
 - Clustering data streams

Clustering data stream methodologies

- Compute and store summaries of past data
- Apply divide-and-conquer strategy
 - Divide stream into chunks, summarise chunks, merge summaries
- Incremental clustering of incoming data (refine clusters)
- Perform micro-clustering as well as macro-clustering
 - Micro-clusters based on hierarchical bottom-up clustering
- Explore multiple time-granularity for cluster evolution
- Divide stream clustering into on- and off-line processes
 - Compute basic summaries of data snapshots online

Challenges for mining dynamics in data streams

- Most stream data is at a pretty low-level or multi-dimensional in nature: needs ML/MD processing
- Analysis requirements
 - Multi-dimensional trends and unusual patterns
 - Capturing important changes at multi-dimensions/levels
 - Fast, real-time detection and response
- Stream (data) cube or stream OLAP: Is this feasible?
 - Can we implement it efficiently?
 - More details in text book

Multi-dimensional stream analysis: Examples

- Analysis of Web click streams
 - Raw data at low levels: seconds, Web page addresses, user IP addresses, IP port numbers, ...
 - Analysts want: changes, trends, unusual patterns, at reasonable levels of details
 - For example: *Average clicking traffic in North America on sports in the last 15 minutes is 40% higher than that in the last 24 hours*
- Analysis of power consumption streams
 - Raw data: power consumption flow for every household, every minute
 - Patterns one may find: *average hourly power consumption surges up 30% for manufacturing companies in Chicago in the last 2 hours today than that of the same day a week ago*

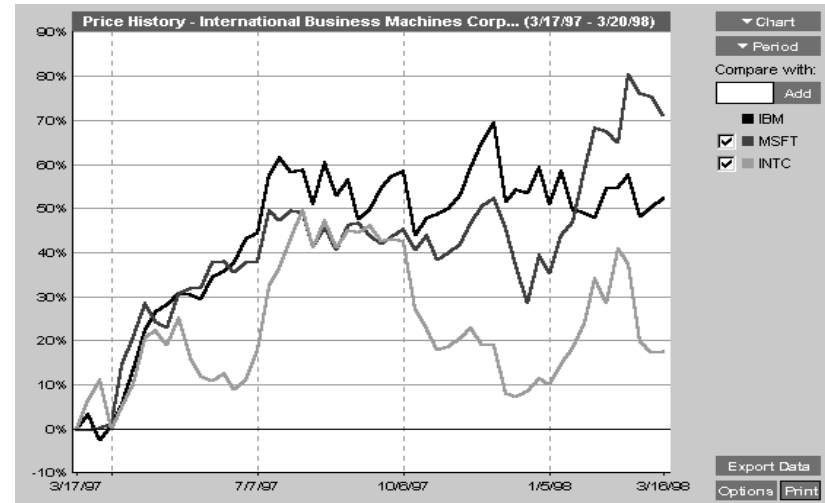
Mining time-series data

- Time-series database

- Consists of sequences of values or events changing with time
- Data is recorded at *regular intervals*
- Characteristic time-series components: Trend, cycle, seasonal, irregular

- Applications

- Financial: stock price, inflation
- Industry: power consumption
- Scientific: experiment results
- Meteorological: precipitation
- Medical treatments



- A time series can be illustrated as a time-series graph which describes a point moving with the passage of time

Categories of time-series movements

- Long-term or trend movements (trend curve) (T)
 - General direction in which a time-series is moving over a long interval of time
- Cyclic movements or cycle variations (C)
 - Long term oscillations about a trend line or curve
 - For example, business cycles, may or may not be periodic
- Seasonal movements or seasonal variations (S)
 - Almost identical patterns that a time series appears to follow during corresponding months of successive years
- Irregular or random movements (I)
- Time series analysis: decomposition of a time-series into these four basic movements
 - Additive Model: $TS = T + C + S + I$
 - Multiplicative Model: $TS = T \times C \times S \times I$

Estimation of trend curve

- The freehand method
 - Fit the curve by looking at the graph
 - Costly and barely reliable for large-scaled data mining
- The least-square method
 - Find the curve minimising the sum of the squares of the deviation of points on the curve from the corresponding data points
- The moving-average method
 - Smooths the data
 - Eliminates cyclic, seasonal and irregular movements
 - Loses the data at the beginning or end of a series
 - Sensitive to outliers (can be reduced by weighted moving average)

$$\frac{y_1 + y_2 + \dots + y_n}{n}, \frac{y_2 + y_3 + \dots + y_{n+1}}{n}, \frac{y_3 + y_4 + \dots + y_{n+2}}{n}, \dots$$

Trend discovery: Estimation of seasonal variations

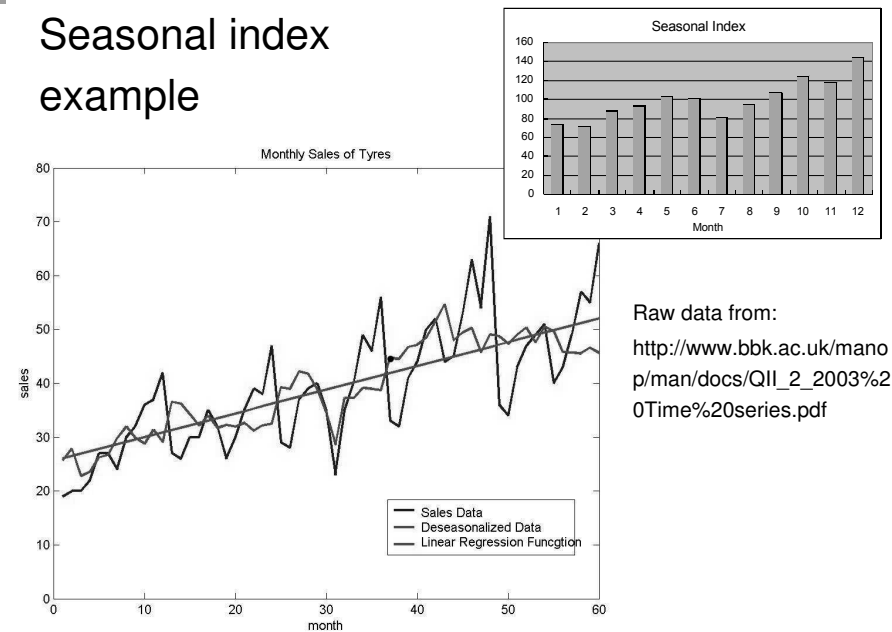
- Seasonal index

- Set of numbers showing the relative values of a variable during the months of the year
- For example, if the sales during October, November, and December are 80%, 120%, and 140% of the average monthly sales for the whole year, respectively, then 80, 120, and 140 are seasonal index numbers for these months

- De-seasonalised data

- Data adjusted for seasonal variations for better trend and cyclic analysis
- Divide the original monthly data by the seasonal index numbers for the corresponding months

Seasonal index example



Trend discovery (2)

- Estimation of cyclic variations

- If (approximate) periodicity of cycles occurs, cyclic index can be constructed in much the same manner as seasonal indices

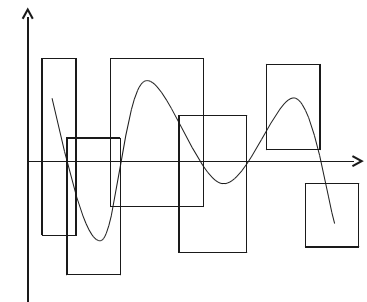
- Estimation of irregular variations

- By adjusting the data for trend, seasonal and cyclic variations

- With the systematic analysis of the trend, cyclic, seasonal, and irregular components, it is possible to make long- or short-term predictions with reasonable quality

Similarity search in time-series analysis

- Normal database query finds exact match
- Similarity search finds data sequences that differ only slightly from the given query sequence
- Two categories of similarity queries
 - *Whole matching*: find a sequence that is similar to the query sequence
 - *Subsequence matching*: find all pairs of similar sequences
- Typical Applications
 - Financial market (similar stocks)
 - Market basket data analysis
 - Scientific databases
 - Medical diagnosis



Analysis of similar time series

