Search and Data Mining: Techniques

Mining Stream, Time-Series, and Sequence Data

Anya Yarygina

Boris Novikov

Introduction

- Mining data streams
  - Remote sensor constantly generating data
  - Massive; temporally ordered; fast changing; potentially infinite
- Mining time-series data
  - Stock market prices
  - Sequences of values obtained over repeated measurements of time
- Mining sequence patterns in transactional databases
  - Market basket analysis in time
  - Sequences of ordered elements or events, recorded with or without concrete notion of time
- Mining sequence patterns in biological data
  - Biological sequences

Reference

Jiawei Han, Micheline Kamber. Data Mining: Concepts and Techniques, 2nd Edition. The Morgan Kaufmann Series in Data Management Systems, 2006

Mining Data Steams

- Data Streams
  - Real-time surveillance systems, communication networks, Internet traffic, on-line transactions in the financial market, electric power grids, remote sensors
  - Massive; temporally ordered; fast changing; potentially infinite
- Mining data streams
  - Single-scan
  - Multilevel
  - Multidimensional

Methodologies for Stream Data Processing and Stream Data Systems

- Trade off between accuracy and storage
- Summaries of stream data which typically can be used to return approximate answers

- Random sampling
- Sliding windows
- Histograms
- Multi-resolution methods
- Sketches
- Randomized algorithms
- DSMS
- Stream query processing

Random Sampling

- Sampling the stream at periodic intervals
- To obtain an unbiased sampling of the data, we need to know the length of the stream in advance

- Reservoir sampling
  - Unbiased random sample of s elements without replacement
  - Idea
    - Maintain a sample of size at least s (reservoir) from which a random sample of size s can be generated
  - Approach
    - Maintain a set of s candidates in reservoir, which form a true random sample of the elements seen to far in the stream
    - Every new element has a certain probability of replacing an old element in reservoir chosen at random (s/N)

Sliding Window

- Make decisions based only on recent data
- Technique
  - At every time t a new data element arrives
  - This element “expires” at time t+w, where w is the window length
- Stocks or sensor networks

Histograms

- Synopsis data structure that can be used to approximate the frequency distribution of element values
- Partitioning techniques
  - Equal-width
  - V-optimal

Multi-Resolution Methods

- Data reduction methods
- Divide-and-conquer strategy
  - Balanced binary tree
  - Hierarchy of micro-clusters
  - Wavelets

Sketches

- U={1,2,…,v}
- A={a1,…,aN}
- Frequency moments of A
  - $F_k = \sum_1^v m^k{}_i$
  - F_0 is number of distinct elements in sequence
  - F_1 is length sequence
  - F_2 is self-join size
- Sketches
  - Approximate F_0, F_1, F_2 in O(log v + log N)
  - Idea
    - Hash every element uniformly at random to either $z_i \in \{-1, +1\}$
    - Maintain a random variable $X = \sum_i m_i z_i$
    - X^2 is a good estimate for F_2

Randomized Algorithms

- Las Vegas algorithm
  - Always returns right answer but the running time vary
- Monte Carlo
  - Bounds on the running time, but may not return the correct result

- Chebyshev inequality

$$P(|X - \mu| > k) \leq \frac{\sigma^2}{k^2}$$

- Chernoff bounds
  - $X_1, \ldots, X_n$ independent Poisson trails
  - X is sum of $X_1, \ldots, X_n$

$$P(X < (1 + \delta)\mu) < e^{-\mu\sigma^2/4}$$

Data Stream Management Systems

- End user
- Query processor
  - One-time queries
  - Continuous queries
- Scratch space

Stream OLAP and Stream Data Cubes

- Tiled time frame model
  - The most recent time is registered at the finest granularity; the more distant time is registered at a coarser granularity
    - Natural tiled frame model
    - Logarithmic tiled frame model
    - Progressive logarithmic frame model
- Critical layers
  - Minimal interest layer
  - Observation layer
- Partial materialization of a stream cube
  - Popular path cubing

Frequent Pattern Mining in Data Streams

- Keep track of only a predefined, limited set of items and itemsets
- Approximate set of answers
- Lossy counting algorithm
  - Input parameters:
    - Minimum support threshold σ
    - Error bound ε
  - Idea:
    - Incoming stream is conceptually divided into buckets of width $w = 1/\varepsilon$
    - N is the current stream length
    - Frequency-list data structure (approximate frequency count f, maximum possible error Δ)
    - Add a new item (from bucket b):
      - In frequency-list: increase frequency count
      - Not in frequency-list: add to the frequency list with Δ=b-1
    - Whenever the bucket boundary is reached: remove from frequency-list entry if $f + \Delta \leq b$
  - Properties:
    - No false negatives
    - False positives are quite “positive”
    - Frequency of frequent item can be underestimated by at most εN

Classification of Dynamic Data Streams

- No multiple scans
- Concept drift

- Hoeffding tree algorithm
- Very Fast Decision Tree
- Concept-adapting Very Fast Decision Tree
- Classifier ensemble approach

Hoeffding Tree Algorithm

- Small sample can often be enough to choose an optimal splitting attribute
- Hoeffding bound
  - True mean of r is at least $\overline{r} - \varepsilon$ with probability $1 - \delta$
  - $\varepsilon = \sqrt{\frac{R^2 \ln(1/\delta)}{2N}}$
- Idea:
  - Input:
    - Sequence of training examples S described by attributes A
    - Accuracy parameter δ
    - Evaluation function $G(A_i)$
  - Algorithm
    - Maximize $G(A_i)$
    - Find smallest number of tuples N for which Hoeffding bound is satisfied
    - If $G(A_a) - G(A_b) > \varepsilon$ then select A_a as the best splitting attribute with confidence $1 - \delta$

Very Fast Decision Tree and Concept-adapting VFDT

Very Fast Decision Tree

- Idea
  - Sliding window
  - Repeatedly apply a traditional classifier to the examples in the sliding window
- Properties
  - Sensitive to window size
  - Expensive

Concept-adapting VFDT

- Sliding window
- Update statistics at the nodes by incrementing the counts associated with new examples and decrementing the counts associated with old ones
- If there is a concept drift some nodes may no longer pass the Hoeffding boung
- Alternate splitting tree will be grown with the new best splitting attribute as the root
- Alternate subtree continue to develop without yet being used for classification
- Once alternate subtree becomes more accurate than the existing one and the old subtree is replaced

Clustering Evolving Data Streams

- Compute and store summaries of past data
- Apply a divide-and-conquer strategy
- Incremental clustering
- of incoming data streams
- Perform micro-clustering as well as macro-clustering
- Explore multiple time granularity for the analysis of cluster evolution
- Divide stream clustering into on-line and off-line processes

- Algorithms
  - STREAM (k-medians-based stream clustering algorithm)
  - CluStream (clustering evolving data streams)

STREAM

- Maintain a consistently good clustering of the sequence seen so far using a small amount of memory and time
- Data stream model of points $X_1, \dots, X_N$ with timestamps $T_1, \dots, T_N$
- STREAM: single pass, constant factor approximation algorithm for the k-median problem
  - Process data streams in buckets of m points
  - Cluster bucket points into k clusters
  - Summarize the bucket information by retaining only information regarding the k centers, with each cluster center being weighted by the number of points assigned
  - Weighted centers are clustered to produce another set of O(k) clusters
  - At every level at most m point are retained

CluStream

- Clustering of evolving data streams based on user specified on-line clustering queries
- On-line component
  - Compute and store summary statistics about the data stream using micro-clusters
  - Incremental maintenance of micro-clusters
- Off-line component
  - Macro-clustering
  - Answer various user questions using the stored summary statistics based on tiled frame model

- Micro-cluster is represented as a clustering feature
  - Set of d-dimensional points $X_1, \dots, X_n$ with timestamps $T_1, \dots, T_n$
  - Micro-cluster as (2d+3) feature $(CF2^x, CF1^x, CF2^t, CF1^t, n)$
    - $CF2^x = \sum_{i=1}^{n} {X_i}^2$
    - $CF1^x = \sum_{i=1}^{n} X_i$
    - $CF2^t = \sum_{i=1}^{n} {X_i}^2$
    - $CF1^t = \sum_{i=1}^{n} X_i$
  - Additive and subtractive properties
- Online step
  - Statistical data collection
    - Maintain q micro-clusters
  - Updating micro-clusters
    - Add point to cluster
    - New cluster
      - Merge old clusters
      - Remove least recent cluster
- Off-line step
  - Macro-clustering
    - Time horizon h
    - Number of desired clusters k
  - Cluster evolution analysis
    - Time horizon h
    - Number of desired clusters k
    - Clock times t_1, t_2

Mining Time-Series Data

- Time-series database
  - Sequences of values or events obtained over repeated measurements of time
  - Stock market analysis, sales forecasting
- Sequence database
  - Sequences of ordered events with or without concrete notion of time
  - Web page traversal, customer shopping transactions
- Trend analysis
- Similarity search

Trend Analysis

- $Y = F(t)$
- Goals of time-series analysis:
  - Modelling time series
  - Forecasting time series

- Trend analysis:
  - Trend or long-term movements
    - General direction in which a time series graph is moving over a long interval of time
    - Trend curve or trend line
      - Weighted moving average method
      - Least squares method
  - Cyclic movements or cyclic variations
    - Long-term oscillations about a trend line which may or may not be periodic
  - Seasonal movements or seasonal variations
    - Seasonal or calendar related
  - Irregular or random movements
    - Sporadic motion of time series due to random or chance events

Trend Analysis

- Time-series modelling
  - Regression analysis
  - Decomposition into basic movements

    $$Y = T * C * S * I$$
  - Adjust seasonal fluctuations
    - Seasonal index (relative values of a variable during the months of a year)
    - Autocorrelation (correlation between I and (i-k) element)
  - Trend of the data
    - Moving average of order n
    - Weighted moving average of order n
    - Freehand method
    - Least squares method
  - Cyclic index
- Time-series forecasting
  - Auto-Regressive Integrated Moving Average (ARIMA)

Similarity Search in Time-Series Analysis

- Find sequences that differ only slightly from the given query sequence
  - Subsequence matching
  - Sequence matching

- Data (dimensionality) reduction and transformation techniques
  - Discrete Fourier transformation
  - Discrete wavelet transformation
  - Singular value decomposition
  - Random projection-based sketch techniques
- Indexing methods for similarity search
  - Multidimensional index using first few Fourier coefficients
  - Broke down sequence into a set of “pieces” and store minimal bounding rectangles
- Similarity search methods
  - Euclidean distance
  - Offset
  - Normalization transformation $x_i' = \frac{x_i - \mu}{\sigma}$
  - Gaps
    - Atomic matching
    - Window stitching
    - Subsequence ordering
  - Parameters
    - Sliding window size
    - Width of envelop for similarity
    - Maximum gap
    - Matching fraction
- Query languages for time sequences
  - Time-sequence query language
  - Shape definition language

Mining Sequence Patterns in Transactional Databases

- Sequential pattern mining
  - Frequently occurring ordered events or subsequences as patterns
  - Item, itemset, event, sequence, sequence database, support

Scalable methods for mining sequential patterns

- Apriori property: every nonempty subsequence of a sequential pattern is a sequential pattern
  - GSP: Generalized Sequential Patterns
    - Candidate generate-and-test
    - Recursive candidate generation from seed set
  - SPADE: Sequential Pattern Discovery using Equivalent classes
    - Candidate generate-and-test
    - Vertical data format (itemset:(sequence_ID, event_ID))
  - PrefixSpan: Prefix-projected sequential pattern growth
    - Pattern growth without candidate generation

Mining Sequence Patterns in Biological Data

- Alignment of biological sequences
  - Pairwise alignment
  - BLAST local alignment algorithm
  - Multiple sequence alignment methods
- Hidden Markov model for biological sequence analysis
  - Markov chain
  - Hidden Markov Model
  - Forward algorithm
  - Viterbi algorithm
  - Baum-Welch algorithm

Outline

- Mining data streams
- Mining time-series data
- Mining sequence patterns in transactional databases
- Mining sequence patterns in biological data