ANU MLSS 2010: Data Mining

Part 3: Application techniques and privacy aspects of data mining
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

• Link mining
  • Common link mining tasks
  • Link based object ranking and object classification
  • Link prediction

• Privacy aspects of data mining
  • Privacy and confidentiality
  • Some scenarios
  • Privacy-preserving data mining

• References and resources
Characteristics of data streams

• Data streams
  • Continuous, ordered, changing, fast, huge amount
  • In a traditional DBMS, data is stored in finite, well-defined and persistent tables

• 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 are required (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
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

DSMS (Data Stream Management System)

Continuous query

Multiple streams

Stream Query Processor

Scratch Space (Main memory and/or Disk)

User/Application

Results

Source: Han and Kamber, DM Book, 2nd Ed. (Copyright © 2006 Elsevier Inc.)
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
- Approximate query answering
  - With bounded memory, it is not always possible to produce exact answers (high quality approximate answers are desired)
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 $\varepsilon$ 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
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*
Link / Network mining

• Heterogeneous, multi-relational data is represented as a graph or network
  • Nodes are objects
    • May have different kinds of objects
    • Objects have attributes
    • Objects may have labels or classes
  • Edges are links
    • May have different kinds of links
    • Links may have attributes
    • Links may be directed, are not required to be binary

• Links represent relationships and interactions between objects - rich content for data mining
What is new for link mining?

• Traditional machine learning and data mining approaches assume:
  • A random sample of homogeneous objects from a single relation

• Real world data sets:
  • Multi-relational, heterogeneous and semi-structured

• Link Mining
  • Newly emerging research area at the intersection of research in social network and link analysis, hypertext and web mining, graph mining, and relational learning
Common link mining tasks

- **Object-Related Tasks**
  - Link-based object ranking
  - Link-based object classification
  - Object clustering (group detection)
  - Object identification (entity resolution)

- **Link-Related Tasks**
  - Link prediction

- **Graph-Related Tasks**
  - Subgraph discovery
  - Graph classification
  - Generative model for graphs
What is a link in link mining?

• Link: relationship among data

• Two kinds of linked networks
  • Homogeneous vs. Heterogeneous

• Homogeneous networks
  • Single object type and single link type
  • Single model social networks (e.g., friends)
  • WWW: a collection of hyper-linked Web pages

• Heterogeneous networks
  • Multiple object and link types
  • Medical network: patients, doctors, disease, contacts, treatments
  • Bibliographic network: publications, authors, venues, affiliations; co-authorship relations, published in/at relations, working at relations
Link-based object ranking (LBR)

• LBR: Exploiting the link structure of a graph to order or prioritize the set of objects within the graph
  • Focused on graphs with single object type and single link type

• This is a primary focus of the link analysis community

• Web information analysis
  • PageRank (Google) and Hits (Hyperlink-Induced Topic Search) are typical LBR approaches

• In social network analysis (SNA), LBR is a core analysis task
  • Objective: rank individuals in terms of “centrality”
  • Rank objects relative to one or more relevant objects in the graph vs. ranks object over time in dynamic graphs
Link-based object classification (LBC)

- Predicting the category of an object based on its attributes, its links and the attributes of linked objects
- **Web**: Predict the category of a web page, based on words that occur on the page, links between pages, anchor text, HTML tags, etc.
- **Citation**: Predict the topic of a paper, based on word occurrence, citations, co-citations
- **Epidemics**: Predict disease type based on characteristics of the patients infected by the disease
- **Communication**: Predict whether a communication contact is by email, phone call or mail
Link prediction

• Predict whether a link exists between two entities, based on attributes and other observed links

• Applications
  • **Web**: predict if there will be a link between two pages
  • **Citation**: predicting if a paper will cite another paper
  • **Epidemics**: predicting who a patient’s contacts are

• Methods
  • Often viewed as a binary classification problem
  • Local conditional probability model, based on structural and attribute features
  • Difficulty: sparseness of existing links
  • Collective prediction, e.g., Markov random field model
Use of labeled and unlabeled data

• In link-based domains, unlabeled data provide three sources of information:
  • Links between unlabeled data allow us to make use of attributes of linked objects
  • Links between labeled data and unlabeled data (training data and test data) help us make more accurate inferences

• Knowledge is power, but knowledge is hidden in massive links
Privacy and confidentiality

• Privacy of individuals
  • Identifying information: Names, addresses, telephone numbers, dates-of-birth, driver licenses, racial/ethnic origin, family histories, political and religious beliefs, trade union memberships, health, sexual orientation, income, ...
  • Some of this information is publicly available, other is not
  • Individuals are happy to share some information with others (to various degrees)

• Confidentiality in organisations
  • Trade secrets, corporate plans, financial status, planned collaborations, ...
  • Collect and store information about many individuals (customers, patients, employees)

• Conflict between individual privacy and information collected by organisations
  • Privacy-preserving data mining and data sharing mainly of importance when applied between organisations (businesses, government agencies)
Protect individual privacy

• Individual items (records) in a database must not be disclosed
  • Not only personal information
  • Confidential information about a corporation
  • For example, transaction records (bank account, credit card, phone call, etc.)

• Disclosing parts of a record might be possible
  • Like name or address only (but if data source is known even this can be problematic)
  • For example, a cancer register, HIV database, etc.

• Remove *identifier* so data cannot be traced to an individual
  • Otherwise data is not private anymore
  • But how can we make sure data can't be traced?
Real world scenarios
(based on slides by Chris Clifton, http://www.cs.purdue.edu/people/clifton)

• Multi-national corporation
  • Wants to mine its data from different countries to get global results
  • Some national laws may prevent sending some data to other countries

• Industry collaboration
  • Industry group wants to find best practices (some might be trade secrets)
  • A business might not be willing to participate out of fear it will be identified as conducting bad practice compared to others

• Analysis of disease outbreaks
  • Government health departments want to analyse such topics
  • Relevant data (patient backgrounds, etc.) held by private health insurers and other organisations (can/should they release such data?)
More real world scenarios (data sharing)

• Data sharing between companies
  • Two pharmaceutical companies are interested in collaborating on the expensive development of new drugs
  • Companies wish to identify how much overlap of confidential research data there is in their databases (but without having to reveal any confidential data to each other)
  • Techniques are needed that allow sharing of large amounts of data in such a way that similar data items are found (and revealed to both companies) while all other data is kept confidential

• Geocoding cancer register addresses
  • Limited resources prohibit the register to invest in an in-house geocoding system
  • Alternative: The register has to send their addresses to an external geocoding service/company (but regulatory framework might prohibit this)
  • Complete trust needed in the capabilities of the external geocoding service to conduct accurate matching, and to properly destroy the register’s address data afterwards
Re-identification

• **L. Sweeney** (Computational Disclosure Control, 2001)
  - Voter registration list for Cambridge (MA, USA) with 54,805 people: 69% were unique on postal code (5-digit ZIP code) and date of birth
  - 87% in whole of population of USA (216 of 248 million) were unique on: ZIP, date of birth and gender!
  - Having these three attributes allows linking with other data sets (quasi-identifying information)

• **R. Chaytor** (Privacy Advisor, SIGIR 2006)
  - A patient living in a celebrity's neighbourhood
  - Statistical data (e.g. from ABS – Australian Bureau of Statistics) says one male, between 30 and 40, has HIV in this neighbourhood (ABS mesh block: approx. 50 households)
  - A journalist offers money in exchange of some patients medical details
  - How much can the patient reveal without disclosing the identity of his/her neighbours?
Goals of privacy-preserving data mining

• Privacy and confidentiality issues normally do not prevent data mining
  • Aim is often summary results (clusters, classes, frequent rules, etc.)
  • Results often do not violate privacy constraints (they contain no identifying information)
  • But, certain rules or classification outcomes might compromise confidentiality
  • But: Certain techniques (e.g. outlier detection) aim to find specific records (fraudulent customers, potential terrorists, etc.)
  • Also, often detailed records are required by data mining algorithms

• The problem is: How to conduct data mining without accessing the identifying data
  • Legislation and regulations might prohibit access to data (especially between organisations or countries)

• Main aim is to develop algorithms to modify the original data in some way, so that private data and private knowledge remain private even after the mining process
Privacy-preserving data mining techniques (1)

• Many approaches to preserve privacy while doing data mining
  • Distributed data: Either horizontally (different records reside in different locations) or vertically (values for different attributes reside in different locations)

• Data modifications and obfuscation
  • Perturbation (changing attribute values, e.g. by specific new values -- mean, average - or randomly)
  • Blocking (replacement of values with for example a '?')
  • Aggregation (merging several values into a coarser category, similar to concept hierarchies)
  • Swapping (interchanging values of individual records)
  • Sampling (only using a portion of the original data for mining)

• Problems: Does this really protect privacy? Still good quality data mining results?
Privacy-preserving data mining techniques (2)

• Data summarisation
  • Only the needed facts are released at a level that prohibits identification of individuals
  • Provide overall data collection statistics
  • Limit functionality of queries to underlying databases (statistical queries)
  • Possible approach: $k$-anonymity (L. Sweeney, 2001): any combination of values appears at least $k$ times

• Problems
  • Can identifying details still be deducted from a series of such queries?
  • Is the information accessible sufficient to perform the desired data mining task?
Privacy-preserving data mining techniques (3)

• Data separation
  • Original data held by data creator or data owner
  • Private data is only given to a trusted third party
  • All communication is done using encryption
  • Only limited release of necessary data
  • Data analysis and mining done by trusted third party

• Problems
  • This approach secures the data sets, but not the potential results!
  • Mining results can still disclose identifying or confidential information
  • Can and will the trusted third party do the analysis?
  • If several parties involved, potential of collusion by two parties

• Privacy-preserving approaches for association rule mining, classification, clustering, etc. have been developed
Secure multi-party computation

• Aim: To calculate a function so that no party learns the values of the other parties, but all learn the final result
  • Assuming semi-honest behaviour: Parties follow the protocol, but they might keep intermediate results

• Example: Simple secure summation protocol (Alan F. Karr, 2005)
  • Consider $K > 2$ cooperating parties (businesses, hospitals, etc.)
  • Aim: to compute $v = \sum_{j=1}^{k} v_j$ so that no party learns other parties $v_j$
  • Step 1: Party 1 generates a large random number $R$, with $R >> v$
  • Step 2: Party 1 sends $(v_1 + R)$ to party 2
  • Step 3: Party 2 adds $v_2$ to $v_1 + R$ and sends $(v_1 + v_2 + R)$ to party 3 (and so on)
  • Step $K+1$: Party $K$ sends $(v_1 + v_2 + ... + v_k + R)$ back to party 1
  • Last step: Party 1 subtracts $R$ and gets final $v$, which it then sends to all other parties
References and resources (1)

• Data mining books:
  • *The Elements of Statistical Learning: Data Mining, Inference and Prediction*, T. Hastie, R. Tibshirani and J. Friedman, 2\textsuperscript{nd} Edition (2009) Springer

• Web resources:
  • www.kdnuggets.com (Email newsletter, courses, jobs, conferences)
  • www.kmining.com (conference calendar, people)
  • www.togaware.com (Graham Williams, Australian Taxation Office)
References and resources (2)

• Open source data mining software:
  • *Rattle* (R based): [www.togaware.com/rattle](http://www.togaware.com/rattle)
    (Graham Williams, Australian Taxation Office)
    (University of Waikato, NZ and Pentaho)
  • *KNIME* (Java based): [www.knime.org](http://www.knime.org)
    (University of Konstanz, Germany)

• Conferences and journals
  • *ACM SIGKDD*: [www.sigkdd.org](http://www.sigkdd.org) (also Explorations news letter)
  • *Springer Data Mining and Knowledge Discovery*: [http://www.springerlink.com/content/100254](http://www.springerlink.com/content/100254)
  • *Springer Knowledge and Information Systems*: [http://springerlink.metapress.com/content/105441/](http://springerlink.metapress.com/content/105441/)
  • *IEEE Transactions on Knowledge and Data Engineering*: [http://www.computer.org/tkde](http://www.computer.org/tkde)