Carnegie Mellon University
15-826 Multimedia Databases & Data Mining
Spring 2017 - C. Faloutsos

Project Description: Dense Subtensor Mining using SQL

Designed by: Kijung Shin (kijungs@cs.cmu.edu)

(updated: 3/22/2017, by C. Faloutsos)

1. Introduction

Multi-aspect data (i.e., tensors) are naturally expressed and stored as a table in relational databases. Can Structured Query Language (SQL), which has been used for simple manipulation of data, also be used for mining patterns and anomalies in multi-aspect data stored in relational databases? Using SQL enables us to handle large-scale data not fitting in memory, easily leveraging recent advancement made in the databases community.

In this project, you will utilize SQL for a task called dense subtensor mining. Specifically, with a teammate, you are to do the following tasks:

- Implement a recent dense-subtensor mining algorithm using SQL,
- Optimize your implementation using indexing methods,
- Evaluate the performance of your implementation using ROC curve,
- Apply the algorithm in real-world data finding interesting patterns and/or anomalies.

2. Introductory Material

2.1. Dense Subtensor Mining

How can we detect dense subtensors, which signal suspicious lock-step behavior, in multi-aspect data? Specifically, how can we measure the suspiciousness of a subtensor and find subtensors maximizing the suspiciousness? The following papers give an answer to these questions.

- **CrossSpot** [http://alexbeutel.com/papers/crossspot-icdm15-paper.pdf]: Axiomatic analysis of suspiciousness measures and an alternating optimization algorithm.
- **M-Zoom** [http://www.cs.cmu.edu/~kijungs/papers/mzoomPKDD2016.pdf]: A near-linear time optimization algorithm with an approximation guarantee.
- **D-Cube** [http://www.cs.cmu.edu/~kijungs/papers/dcubeWSDM2017.pdf]: A space-efficient optimization algorithm for large-scale data not fitting in memory.

2.2. Graph Mining Using SQL

Are relational databases and SQL powerful enough to analyze large-scale graph data? How can complicated graph algorithms be expressed in SQL? How can we optimize SQL implementations of graph algorithms? The following papers give an answer to these questions.

- **Vertexica** [https://arxiv.org/pdf/1412.5263v1.pdf]: Optimized SQL implementation of vertex-centric graph processing on modern column-oriented databases
- **GBase** [http://datalab.snu.ac.kr/~ukang/papers/GbaseVLDBJ2012.pdf]: SQL implementations of graph queries, such as k-step neighbors, k-step egonets, and single source shortest distances.
- **Pegasus:** [https://www.cs.cmu.edu/~christos/PUBLICATIONS/icdm09-pegasus.pdf] : how to use hadoop to analyze large graphs. All graph operations eventually become generalized matrix-vector multiplication operations, which can be implemented as joins, in SQL. (new, 2/12/17)

The above 6 papers are fine for your survey – feel free to substitute any/all of them with papers you deem relevant; you may also survey **more** than 3 papers per person.

Check an earlier class project, to see how SQL can handle graph mining operations (diameter, singular/eigenvalues, degree distribution, etc).

- **GraphMiner** [http://www.cs.cmu.edu/~christos/courses/826.S16/project-default-graphs/graphminer.tar.gz]: SQL implementations of major graph mining algorithms, including PageRank, Eigenvalue Computation, and Triangle Count. The write-up is in the ./doc directory

3. Datasets

Multi-aspect data are everywhere! The following is a partial list of publicly available multi-aspect datasets. For your convenience, we have collected them at http://www.cs.cmu.edu/~christos/courses/826-resources/DATA-SETS-graphs/ (new 3/19/17)

- **Amazon Review**
  - 4-way tensor: (user, item, timestamp, rating)
  - source: http://jmcauley.ucsd.edu/data/amazon/]
- **Yelp Review**
  - 4-way tensor: (user, business, timestamp, rating)
  - source: https://www.yelp.com/dataset_challenge/dataset]
- **English Wikipedia Revision History**
  - 3-way tensor: (user, page, timestamp)
  - source: https://dumps.wikimedia.org/
- **DARPA TCP Dump**
  - 3-way tensor: (source ip, destination ip, timestamp)
  - source: https://www.ll.mit.edu/ideval/data/1998data.html
- **AirForce TCP Dump**

- 7-way tensor: see the data description for details
- source: https://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

4. Project Tasks

T1. **Literature Survey**: Complete a literature survey: at least 3 paper reviews per team member, from the introductory papers above (Section 2). Paper reviews should consist of

a) the **problem** definition that the paper is addressing

b) a summary of the **main idea** of the paper (in your own words - cutting-and-pasting text from the paper or any other source, is plagiarism)

c) why, or why not, this paper is **useful** for your project (new 2/12/17)

d) list of **shortcomings**, that you think that future research could address.

T2. **SQL Implementation**: Implement D-Cube using SQL. Specifically,

- **Given:** (1) multi-aspect data: ***R***, (2) number of dimension attributes: ***N***, (3) number of dense blocks we aim to find: ***k***, (4) density measure: $\boldsymbol{\rho}$, and (5) a dimension-selection policy
- **Output**: (1) compute the k dense blocks in the given multi-aspect data and store them in a table; and (2) return the densities of the k dense blocks

[Note]

1. Do not assume that neither given data nor metadata (attribute-value masses, etc.) fit in memory. That is, both given data and metadata should be stored and manipulated in a database by SQL.

2. Your implementation should support three density measures (i.e., arithmetic average degree, geometric average degree, and suspiciousness) and two dimension-selection policies (i.e., maximum density first and maximum cardinality first)

2'. *Single-member* teams should do *only* the 'geometric average degree' density measure, and *only* the 'maximum cardinality first' dimension-selection policy. (new 3/8/2017)

3. All teams (single/double-member ones) should present results on the '*DARPA TCP dump*' dataset, only (new 3/12/2017).

4. IMPORTANT (new 3/22/2017): make sure your code includes a 'makefile', so that 'make' should run your code on the darpa.csv file

4.1. Bucketize time (say, by hour, or day);

4.2. you may binarize the count: '0' remains '0'; any count >0 becomes '1'. That is, if source 128.1.1.1 sent 5 packets to destination 127.2.2.2, you may treat it as '1'.

4.3. DELIVERABLES:

- upload your code (tar-file) on blackboard; and
- a hard copy of your phase2 report (plus the graded phase1 report)
- no need to give hard copy of your code

4.4 In your report, make sure you list the **unit-tests** you tried

T3. **Optimization using Indexing Methods**: Experiment with various indexing options (*create index* in postgres) to find the fastest setting for your implementation. Describe the settings you tried, the one you think is best, and the wall-clock times that support your opinion. Also, compare two possible implementations: (1) copying the input relation (denoted by **R** in the paper) for initializing each block (denoted by **B** in the paper) and actually removing tuples from the copied relation; and (2) adding a column to the input relation (denoted by **R** in the paper) to mark removed tuples.

T4. **Evaluation using ROC Curve**: Apply your implementation of D-Cube to AirForce TCP Dump and DARPA TCP Dump (please use preprocessed versions) for detecting network intrusion. (1) Report the the size, mass, and density of the detected blocks in each dataset with $\rho=\rho ari$, $k=5$, and Maximum density policy. (2) Explain whether the detected blocks indicate specific types of network attacks. (3) Draw a ROC curve and compute AUC for each dataset with $\rho=\rho ari$ and Maximum density policy. (4) Observe how AUC changes depending on density measures and dimension-selection policies.

[Note] The ROC Curve plots *False Positive Rate* (number of benign connections within k dense blocks / total number of benign connections) versus *True Positive Rate* (number of network attacks within k dense blocks/ total number of network attacks) at various k values in $\{1,2\ldots,20\}$. Use linear interpolation between data points to calculate the AUC. Note that this way of drawing a ROC Curve and computing AUC is different from that in the paper.

T5. **Anomaly Detection in Real-world Data**: Apply your implementation of D-Cube to the real-world multi-aspect datasets above (Section 3). (1) Report the size, mass, and density of detected blocks in each dataset with $k=5$ and Maximum density policy. Regarding density measures, use ρgeo for English Wikipedia Revision History and ρari for the other datasets. (2) Provide a justification why you believe the detected blocks do or do not indicate interesting anomalies.

5. Phases/deliverables:

Phase 1: Task 1 (literature survey)
Phase 2: Task 2 (SQL implementation)
Phase 3: all other tasks.

6. Appendix: Dataset descriptions

6.1. Amazon Rating Dataset (amazon.csv)

- **Description:** Ratings for Android apps provided by Amazon. We anonymized the original dataset and injected anomalies in the original dataset.
- **Source:** http://jmcauley.ucsd.edu/data/amazon/
- **Format:** Each record in '*amazon.csv*' corresponds to a rating. Each record consists of the following four dimension attributes, which are separated by commas:
    - user_id: a unique identifier for users
    - app_id: a unique identifier for apps
    - time_in_hours: mapped to integers for anonymization
    - number_of_stars: an integer between 1 to 5

6.2. Yelp Rating Dataset (yelp.csv)

- **Description:** Ratings for businesses provided by Yelp. We anonymized the original dataset and injected anomalies in the original dataset.
- **Source:** https://www.yelp.com/dataset_challenge/dataset
- **Format:** Each record in '*yelp.txt*' corresponds to a rating. Each record consists of the following four dimension attributes, which are separated by commas:
    - user_id: a unique identifier for users
    - business_id: a unique identifier for apps
    - time_in_hours: mapped to integers for anonymization
    - number_of_stars: an integer between 1 to 5

6.3. DARPA TCP Dump (darpa.csv)

- **Description:** A TCP dump provided by MIT Lincoln Laboratory. We extracted source IPs, destination IPs, and timestamps from the original dataset. Labels, which indicate whether each connection is benign or malicious, are also included in the original dataset.
- **Source:** https://www.ll.mit.edu/ideval/data/1999data.html
- **Format:** Each record in '*darpa.csv*' corresponds to a TCP connection. Each record consists of the following three dimension attributes, which are separated by commas:
    - source_ip
    - destination_ip
    - time_in_minutes
- **Labels**: Labels, which indicate whether each connection is benign or malicious, are provided in '*darpa_with_label.csv*'. The value '-' indicate that the corresponding connection is benign, and the other values indicate a type of network attacks. The description of types of network attacks are available at https://www.ll.mit.edu/ideval/docs/attackDB.html.

6.4. AirForce TCP Dump (airforce.csv)

- **Description:** A TCP dump used for KDD Cup 1999. We extracted 7 dimension attributes from the original dataset. Labels, which indicate whether each connection is benign or malicious, are also included in the original dataset.
- **Source:** https://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html
- **Format:** Each record in '*airforce.csv*' corresponds to a TCP connection. Each record consists of the following seven dimension attributes, which are separated by commas:
    - protocol: type of protocol (e.g. tcp and udp)
    - service: network service on destination (e.g., http and telnet)
    - src bytes: number of data bytes sent from source to destination
    - dst bytes: number of data bytes sent from destination to source
    - flag: normal or error status
    - host count: number of connections made to the same host in the past two seconds
    - srv count: number of connections made to the same service in the past two seconds
- **Labels**: Labels, which indicate whether each connection is benign or malicious, are provided in '*airforce_with_label.csv*'. The value 'normal.' indicate that the corresponding connection is benign, and the other values indicate a type of network attacks.

6.5. English Wikipedia Revision History (wiki.csv)

- **Description:** A revision history in English Wikipedia (https://en.wikipedia.org/) . We extracted history in January 2016 from the original dataset.
- **Source:** : https://dumps.wikimedia.org/
- **Format:** Each record in '*wiki.csv*' corresponds to a revision. Each record consists of the following dimension attributes, which are separated by commas:
    - user_name: users who revised Wikipedia pages
    - page: titles of pages which are revised
    - time_in_hours