CompSci 516
Database Systems

Lecture 26
Data Mining
And Data Cube

Instructor: Sudeepa Roy
Optional Reading Material

1. [RG]: Chapter 26

2. “Fast Algorithms for Mining Association Rules”
   Agrawal and Srikant, VLDB 1994

28,871 citations on Google Scholar in April 2022!
• 25,426 in November 2019
• 23,863 in November 2018
• 23,038 in November 2017
• 20,610 in November 2016
• 19,496 in April 2016

One of the most cited papers in CS!

• Acknowledgement:
  The following slides have been prepared adapting the slides provided by the authors of [RG] and using several presentations from the internet
Three Typical Complementary Trends in Data Analysis

• **Data Warehousing (DW):**
  – Consolidate / integrate data from many sources in one large repository
  – Loading, periodic synchronization of replicas

• **OLAP:**
  – Complex SQL queries and views.
  – Queries based on spreadsheet-style operations and “multidimensional” view of data.
  – Interactive and “online” queries.

• **Data Mining:**
  – Exploratory search for interesting trends and anomalies
Four Main Steps in KD and DM (KDD)

• **Data Selection**
  – Identify target subset of data and attributes of interest

• **Data Cleaning**
  – Remove noise and outliers, unify units, create new fields, use denormalization if needed

• **Data Mining**
  – extract interesting patterns

• **Evaluation**
  – present the patterns to the end users in a suitable form, e.g., through visualization

Remember HW1!
Several DM/KD (Research) Problems

- Discovery of causal rules
- Learning of logical definitions
- Fitting of functions to data
- Clustering
- Classification
- Inferring functional dependencies from data
- Finding “usefulness” or “interestingness” of a rule
Example Data Mining:
Mining Association Rules by Apriori
Mining Association Rules

• Retailers collect and store massive amounts of sales data
  – transaction date and list of items

• Association rules:
  – e.g. 98% customers who purchase “tires” and “auto accessories” also get “automotive services” done
  – Customers who buy mustard and ketchup also buy burgers
  – Goal: find these rules from just transactional data (transaction id + list of items)
Applications

• Can be used for
  – marketing program and strategies
  – cross-marketing (mass e-mail, webpages)
  – catalog design
  – add-on sales
  – store layout
  – customer segmentation
Notations

• Items $I = \{ i_1, i_2, \ldots, i_m \}$
• $D$ : a set of transactions
• Each transaction $T \subseteq I$
  – has an identifier $TID$
• Association Rule
  – $X \rightarrow Y$ (not Functional Dependency!)
  – $X, Y \subset I$
  – $X \cap Y = \emptyset$
Confidence and Support

• Association rule $X \rightarrow Y$

• Confidence $c = \frac{|\text{Tr. with } X \text{ and } Y|}{|\text{Tr. with } X|}$
  – $c\%$ of transactions in $D$ that contain $X$ also contain $Y$

• Support $s = \frac{|\text{Tr. with } X \text{ and } Y|}{|\text{all Tr.}|}$
  – $s\%$ of transactions in $D$ contain $X$ and $Y$. 
## Support Example

<table>
<thead>
<tr>
<th>TID</th>
<th>Cereal</th>
<th>Beer</th>
<th>Bread</th>
<th>Bananas</th>
<th>Milk</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>2</td>
<td>X</td>
<td></td>
<td>X</td>
<td>X X</td>
<td>X X</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>4</td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>6</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>

- Support(Cereal)
  - 4/8 = 0.5
- Support(Cereal → Milk)
  - 3/8 = 0.375
# Confidence Example

<table>
<thead>
<tr>
<th>TID</th>
<th>Cereal</th>
<th>Beer</th>
<th>Bread</th>
<th>Bananas</th>
<th>Milk</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>2</td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>4</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>6</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>

- **Confidence(Cereal → Milk)**
  - $\frac{3}{4} = .75$
- **Confidence(Bananas → Bread)**
  - $\frac{1}{3} = .33333...$
X → Y is not a Functional Dependency

For functional dependencies

• F.D. = two tuples with the same value of X must have the same value of Y
  – X → Y => XZ → Y (concatenation)
  – X → Y, Y → Z => X → Z (transitivity)

For association rules

• X → A does not mean XY → A
  – May not have the minimum support
  – Assume one transaction {AX}

• X → A and A → Z do not mean X → Z
  – May not have the minimum confidence
  – Assume two transactions {XA}, {AZ}
Problem Definition

• **Input**
  – a set of transactions $D$
    • Can be in any form – a file, relational table, etc.
  – min support threshold (minsup)
  – min confidence threshold (minconf)

• **Goal: generate all association rules that have**
  – support $\geq$ minsup and
  – confidence $\geq$ minconf
Decomposition into two subproblems

• **1. Apriori**
  – for finding “large” itemsets with support $\geq \minsup$
  – all other itemsets are “small”

• **2. Then use another algorithm to find rules $X \rightarrow Y$ such that**
  – Both itemsets $X \cup Y$ and $X$ are large
  – $X \rightarrow Y$ has confidence $\geq \minconf$

• **Paper focuses on subproblem 1**
  – if support is low, confidence may not say much
  – subproblem 2 in full version of the paper
Basic Ideas

• Q. Which itemset can possibly have larger support: ABCD or AB
  – i.e., when one is a subset of the other in terms of the set of transactions containing them?
Basic Ideas

• Q. Which itemset can possibly have larger support: ABCD or AB
  – i.e., when one is a subset of the other in terms of the set of transactions containing them?

• Ans: AB
  – any subset of a large itemset must be large
  – So if AB is small, no need to investigate ABC, ABCD etc.
Apriori Algo Overview

• Start with individual (singleton) items \{A\}, \{B\}, ...

• In subsequent passes, extend the “large itemsets” of the previous pass as “seed”

• Generate new potentially large itemsets (candidate itemsets)
  – E.g., if \{AB\} \{AC\} are two large itemsets of size 2, a candidate itemset for size 3 is \{ABC\} (different last item in the otherwise same sequence)

• Then count their actual support from the data

• At the end of the pass, determine which of the candidate itemsets are actually large
  – becomes seed for the next pass

• Continue until no new large itemsets are found
Example: Generation of potentially large itemsets in next round (1/3)

Actual large items of size 3

• \{1,2,3\}
• \{1,2,4\}
• \{1,3,4\}
• \{1,3,5\}
• \{2,3,4\}

Potential large items of size 4

• \{1,2,3,4\}

“Join” L3 with L3
Such that all except one items are the same
Example: Generation of potentially large itemsets in next round (2/3)

Actual large items of size 3
- \{1,2,3\}
- \{1,2,4\}
- \{1,3,4\}
- \{1,3,5\}
- \{2,3,4\}

Potential large items of size 4
- \{1,2,3,4\}
- \{1,3,4,5\}

“Join” \(L_3\) with \(L_3\)
Such that all except one items are the same.
Example: Generation of potentially large itemsets in next round (3/3)

- Actual large items of size 3:
  - \{1,2,3\}
  - \{1,2,4\}
  - \{1,3,4\}
  - \{1,3,5\}
  - \{2,3,4\}

- Potential large items of size 4:
  - \{1,2,3,4\}
  - \{1,3,4,5\}

- Prune candidates further:
  Remove itemsets that can’t have the required support because there is a subset in it which is not an actual large itemset.

  No \{1,4,5\} exists in \(L_3\)
  Rules out \{1, 3, 4, 5\}

Next verifies that \{1, 2, 3, 4\} indeed has enough support looking into the data.
Data Warehousing, OLAP, and Data Cube

- [RG] – Chapter 25
  - Optional Reading
  - Harinarayan-Rajaraman-Ullman, SIGMOD 1996 “Implementing data cubes efficiently”
Warehousing

• Organizations analyze current and historical data from all parts of an enterprise by complex methods
  – to identify useful patterns
  – to support business strategies

• Growing industry: $8 billion way back in 1998

• Data warehouse vendor like Teradata
  • big “Petabyte scale” customers
  • Apple, Walmart (2008-2.5PB), eBay (2013-primary DW 9.2 PB, other big data 40PB, single table with 1 trillion rows), Verizon, AT&T, Bank of America
  • supports data into and out of Hadoop


Ack: Slide by Prof. Shivnath Babu
Star Schema and Snowflake Schema

- Reflects multi-dimensional views of data
- Single fact table, multiple dimension tables (foreign keys)

- Dimensional hierarchy is explicitly represented
- (+) Dimension tables easier to maintain
- (-) Need additional joins
Data Cube: Intuition

For analyzing sales trends:

Find total sales for all Models, Models by Year, Color by Year...

More complex to do these with GROUP-BY

SELECT 'ALL', 'ALL', 'ALL', sum(units)
FROM Sales
UNION
SELECT 'ALL', 'ALL', Color, sum(units)
FROM Sales
GROUP BY Color
UNION
SELECT 'ALL', Year, 'ALL', sum(units)
FROM Sales
GROUP BY Year
UNION
SELECT Model, Year, 'ALL', sum(units)
FROM Sales
GROUP BY Model, Year
UNION
...

- How many sub-queries? = 8
- How many sub-queries for 8 attributes? $2^8 : \text{SLOW!}$
Data Cube Syntax

Find total sales for all Models, Models by Year, Color by Year...

- Postgres (syntax varies with DB systems)

```
SELECT Model, Year, Color, sum(units) as s
FROM Sales
GROUP BY CUBE(Model, Year, Color)
```

In practice, almost as fast as one simple GROUP-BY, and not 8 * GROUP BY
Uses several optimizations in the implementation
Announcements
Announcements (4/12, Tues)

• Final: 4/27 (Wed), 9 am -12 noon, in class
  – Closed book/notes
  – *Comprehensive*, but likely to have more emphasis on material after midterm
  – Everything covered in Lectures included - ask questions in OH and Ed!
  – No virtual exams, no make-up exams (see final exam rules at Duke)

• See project posts on Ed for Video and grading
  – Any difficulty, issues, or questions – please reach out now
  – Report and video due by Friday 4/15 – noon (extra time)

• Please fill out course evaluations! (see Ed post)
Summary!
Take-Aways

• DBMS Basics

• DBMS Internals

• Overview of Research Areas

• Hands-on Experience in DB systems
DB Systems

• Traditional DBMS
  – PostGres, SQL

• Large-scale Data Processing Systems
  – Spark/Scala

• New DBMS/NOSQL
  – MongoDB

• In addition
  – XML, JSON, JDBC / psycopg2, Python/Java
DB Basics

• SQL
• RA/Logical Plans
• RC
• Recursion in SQL / Datalog
  – Why we needed each of these languages
• Normal Forms
DB Internals and Algorithms

• Storage
• Indexing
• Operator Algorithms
  – External Sort
  – Join Algorithms
• Cost-based Query Optimization
• Transactions
  – Concurrency Control
  – Recovery
Large-scale Processing and Other topics

- Parallel DBMS
- Distributed DBMS
- Map Reduce
- NOSQL
- Data Mining / OLAP

There are a huge number of other topics in database research and applications that we could not cover.

- Hope some of you will further explore Database Systems/Data Management/Data Analysis/Big Data as a researcher or practitioner!