Hasso Plattner Institut

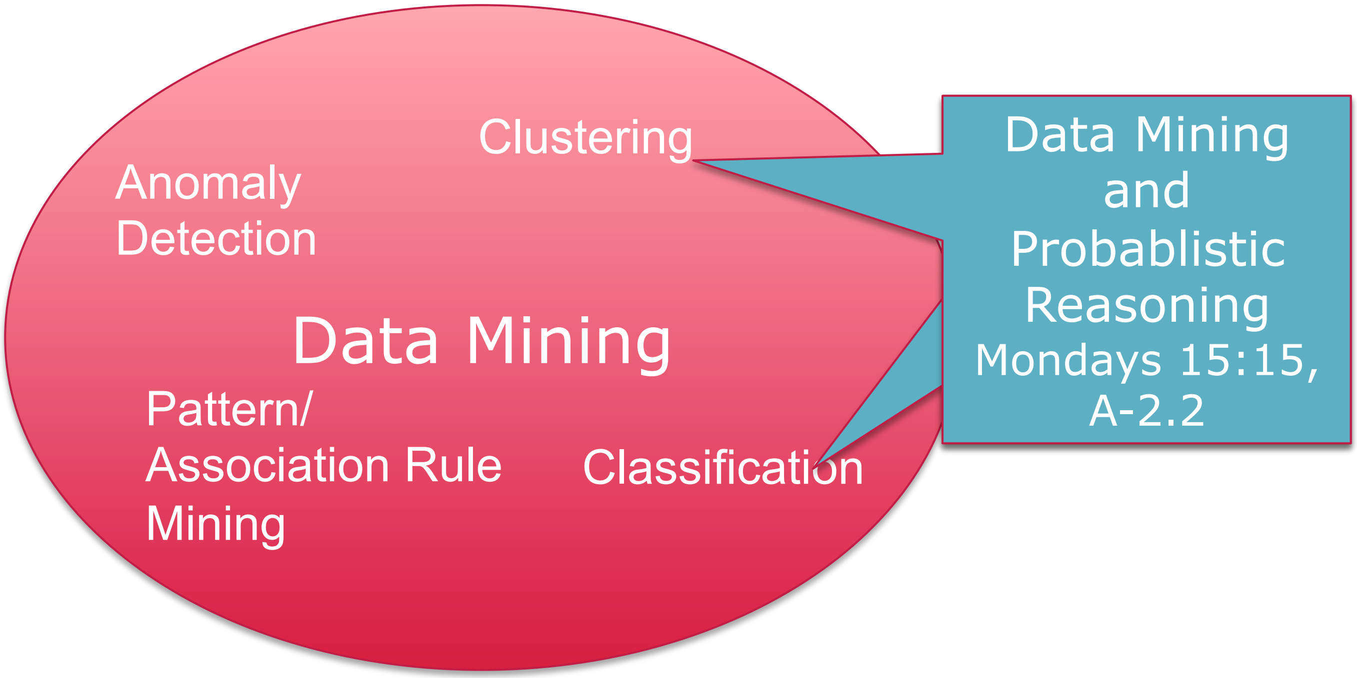
IT Systems Engineering | Universität Potsdam

Introduction to Algorithms for Pattern Mining

Ziawasch Abedjan, Prof. Felix Naumann

Pattern Mining

- Data Mining: "…extract knowledge from a data set in human-understandable structure…"
- Pattern: frequently occurring event or item combinations



Pattern Mining Applications

- Shopping basket analysis
  - Which products are likely to be bought together?

- Web mining
  - Web content mining, web structure mining, web usage mining
- Software bug mining
  - Identify copy and paste code for bug isolation
  - Extract application specific programming rules
- Mining data streams
- Mining multimedia data

Frequent Pattern Mining

TID	transaction
1	bread, milk, tea
2	beer, diaper, bread
3	beer, diaper, bread, milk
4	flour, milk, bread
5	beer

- Frequent pattern
  - holding **support** 25%
  - {milk},{bread}, {beer}, {diaper}
  - {milk, bread}...
- Maximal frequent pattern
  - no proper super-itemset is frequent
  - {milk, bread}, {beer, diaper, bread}
- Closed frequent pattern
  - no proper super-itemset has the same support
  - {milk, bread}, {beer}, {beer, diaper, bread}

Association Rules

- Association Rules
  - For each frequent itemset a generate rules: $l \rightarrow a - l$ where $l \subset a, l \neq \emptyset$
  - Output rules with minimum **confidence** $\mathrm{conf}(l \rightarrow a - l) = \dfrac{\sup(a)}{\sup(l)}$
- Example
  - holding **confidence** 60%
  - Positive Rules
    - {beer} -> {diaper}, 100%
    - {bread} -> {milk}, 75%
  - Negative Rules
    - {tea} -> NOT {coffee}
- Correlation Coefficient, Lift, ...

TID	transaction
1	bread, milk, tea
2	beer, diaper, bread
3	beer, diaper, bread, milk
4	flour, milk, bread
5	beer

FP Mining Algorithms

- Naive approach: scan transaction table for **each** combination for retrieving its support → 2^n scans
- Pruning by the intuition "all subsets of a frequent pattern must also be frequent"
  - 1. Extract all existing relevant itemset frequencies holding minimum support
  - 2. Discover relationships
- Mining Algorithms:
  - Apriori [vldb94]
  - FP-Growth [sigmod00]
  - Eclat [tkde00]

Apriori [agrawal93]

- Bottom-Up approach with multiple passes
- Precondition: all itemsets are sorted lexicographically
- Process:
  - Identify frequent items (1-itemsets)
  - Generate k+1-candidates by combining frequent k-itemsets that have the first k-1 items in common
  - Prune candidates with non-frequent subsets
  - Verify remaining k-candidates

Apriori Example

TID	transaction
1	bread, milk, tea
2	beer, diaper, bread
3	beer, diaper, bread, milk
4	flour, milk, bread
5	beer

- minimum support = 25%

1. Pass:
   - {bread}, {milk}, {beer}, {diaper}
2. Pass: Combine all 1-frequent-itemsets
   - Candidates: {bread, milk}, {beer, bread}, {bread, diaper}, {beer, milk},…
   - After scan: {bread, milk}, {beer, bread}, {bread, diaper}, {beer, diaper}
3. Pass: Combine all 2-frequent-itemsets that have the first item in common
   - Candidates: {bread, milk, diaper}, {beer, bread, diaper}
   - Prune {bread, diaper, milk} because {diaper, milk} is not frequent

Challenges

- Efficient generation of negative association rules
  - Needs tracking non-frequent items as well
- Considering Multi-set semantics
- Non-redundant parallelization

Grading process

- 3 LP
- groups of two (limited to 3 groups)
- Grading
  - Implementation of one algorithm, one use case and, one extension
  - 2 presentations
    - Paper presentation and first algorithm evaluations
    - Use case and extension evaluation
  - 6 pages evaluation report

Topics

- Algorithms:
  - AprioriTID, FPGrowth, Eclat
- Suggested extensions
  - Quantitative association rules [sigmod96], negative associations [tois04], high utility itemsets[kdd10], …
  - Efficiency or scalability boost (paralellizing)
- Suggested data sources/ use cases
  - DBpedia (any other linked data resource) [smer11]
  - www.data-mining-cup.de
  - Source code of large projects
  - www.data.gov

Application

- Send mail to ziawasch.abedjan@hpi.uni-potsdam.de
- Subject [APM Seminar]
- Deadline: April 13th
- Notification: April 14th

- Limited to 6 participants = 3 teams
  - Random selection if more applicants
- Send ranked wishes on algorithms
  - You may also already propose extension and use case
  - You may include desired teammate (Both should write an e-mail)

Time Schedule

- April 10th: first seminar, topic presentation
- April 13th: **application deadline**
- April 14th: notification
- April 17th: mandatory consulting
- April 24th: mandatory consulting
- May 1st: workers unite
- May 8th: mandatory consulting
- May 15th**: intermediate presentation**
- May 22nd: mandatory consulting
- ...
- July 10th: **final presentation**
- July 14th: **short paper deadline**

References

- [vldb94] R. Agrawal & R. Srikant, fast algorithms for mining association rules
- [vldb95] J. Han& Y. Fu, Mining multiple-level association rules in large data bases
- [sigmod96] R. Srikant & R. Agrawal, Mining quantitative association rules in large relational tables
- [sigmod00] J. Han & J. Pei & Y. Yin, Mining frequent patterns without candidate generation
- [tkde00] M. J. Zaki, Scalable algorithms for association mining
- [smer11] Z. Abedjan, F. Naumann, Context and target configurations for mining RDF data
- [kdd10] V. Tseng & C. Wu & B. Shie & P. S. Yu, UP-Growth: an efficient algorithm for high utility itemset mining
- [tois04] X. Wu & C. Zhang & S. Zhang, Efficient mining of both positive and negative association rules