Data Mining Projects for MTAT.03.183


Jaak Vilo, vilo @ ut.ee

1-2-3-4 people project group. Everyone must contribute! Project = about 40h (1 full working week) full of effort from every group member. That is, for 3 person group, the overall effort is about one man-month. Bigger groups are expected to deliver more!

Project should be submitted with a proper final project report – title, authors&affiliations, abstract, results, methods, implementation, discussion, references! Additionally, project presentation as a deck of slides (Powerpoint/OO/etc). Each project shall have an opportunity to present briefly their main results. Be prepared for a 5-minute well-prepared pitch talk for selling your problem, why it is important what did you achieve, and main interesting findings/results!

Project deadline: December 15 (Thu), at 23:59.
Every missed day will reduce 25% of maximal points.

You are free to propose and implement your own project of similar complexity. Too simple tasks will be punished in grading ☹. But also too broad and complex tasks will suffer due incomplete solutions that you can offer and hence inadequate reporting. Therefore, focus your project to some specific well-designed question and examples. Technical implementations are favoured. In some cases prior literature search should be conducted. If literature based, then there should be a serious effort on write-up of the report.

Grading (i.e. nr of points awarded) will be based on the clear statement of the question/objective of the project; it’s technical implementation and coverage; and quality of the report.

Each group will work independently from other groups, especially in cases where the same topic has been chosen by two or more groups.

Groups should be formed and the selected topic and group members should be announced at the course mailing list. Deadline for forming a group and selecting a topic is by next Thursday, Nov 17th. In case of a self-chosen subject, please first consult with me over email, providing similar short task description. And only afterwards send it to the mailing list.
Example project descriptions

Data Mining Projects for MTAT.03.183

1. Comparison of different association rule algorithms
2. Study and comparison of different episode mining rule discovery algorithms
3. Machine learning and optimisation for real estate price estimation
4. Mining and visualising of student flows through the academic career. (graph, visualisation)
5. Condition-specific graph mining
6. Condition-specific graph mining – machine learning approach
7. Study of linear regression using median square error (as opposed to mean square error), evolutionary approaches
8. Perform a computational experimentation on different interestingness measures for association rules, looking at their pairwise correlation, as well as stability under small changes in the data
9. Design and implement a system for automatic descriptive characterisation of data
10. Automatic concept hierarchies
11. Automatic analysis of US census data
12. Scalable K-medoids using triangle inequality
13. Study and visualiser of different clustering methods for density-based clustering
14. Density based clustering
15. Test and improve on DBSCAN. Consider examples where DBSCAN does not perform well out of the box. Propose some ideas about how to possibly modify the method. Demonstrate results
16. Extend K-Means into a density based clustering
17. Density based clustering on different densities
18. SPEXS – finding frequent patterns in text
19. ML Gallery builder
20. Random Forest – good or bad?
1. Comparison of different association rule algorithms

Perform a comparison of different association rule mining algorithms and tools for scalability and performance under different characteristics of underlying data (e.g. generate data with different characteristics and then perform the comparison of the algorithms).

2. Study and comparison of different episode mining rule discovery algorithms

Frequent itemset mining takes as input so called “shopping basket” and tries to find itemsets that are frequent in many “baskets”.

Episode rules are applied on event streams, i.e. (e1,t1),(e2,t2),...(en,tn).

Time can vary. Goal is to find such sets of events that can occur frequently in some sliding windows, say for example, in 100 second intervals. Events can be organised in some partial order, e.g. e1 followed by e2 (e1,e2 are here event types); e1 and e2 in any order; e1 or e2, but nor necessarily both.

Design and implement a programme for mining such frequent episodes.

Focus on finding regions or time intervals that contain many events in short time window, or many repetitions of a small set of events in some time-frame.

Perform a comparison of different episode rule mining algorithms and tools for scalability and performance under different characteristics of underlying data (e.g. generate data with different characteristics and then perform the comparison of the algorithms).

3. Machine learning and optimisation for real estate price estimation

We have Estonian real estate market data about sales. Ultimate goal is to understand how prices depend on location, different characteristics, etc.

Initial goal is to perform the preprocessing and descriptive characterisation of data.

Later there is possibility to continue on with the project and perform prediction of real estate prices.
Supervisors: Kristjan Korjus and Jaak Vilo

4. **Mining and visualising of student flows through the academic career. (graph, visualisation)**

Task – to perform analysis of student progression through the studies.

(We need to ask for appropriate data set – co-supervised with Jaak Vilo, dependent on data availability)

5. **Condition-specific graph mining**

Many data sets are expressed as graphs or networks (of people, gene-gene pairs, object-attribute pairs, etc). But the relationships in between the nodes is usually binary (present or not) or weighted (fixed weight). However, the relationships can be dynamic and depend on the condition or situation.

Your task is to propose some mining of this condition-specific graph, where for each pair of nodes several weights that are condition-specific, are provided. To simplify the analysis imagine the similarity is expressed as the rank of the similarity. Then A-B can be most similar to each other (rank 1) or most different (maximal rank).

E.g

A-B  c1:5  c2:100  c3:1  c4:16  ...
A-C  c1:8  c2:10  c3:2  c4:17  ...
...
X-Y  c1:89  c2:56  ...  cc:100

Your task is to identify subnetworks that are condition specific. E.g. some small subnetwork based on pairwise similarity under certain conditions (which?) versus another subnetwork (partially overlapping?) under different conditions.

You can use this as ranking of each other under different conditions. I would go studying with these and these guys, but I would go partying with these and these...

Co-supervised: Jaak Vilo

6. **Condition-specific graph mining – machine learning approach**

Can you predict gene-gene similarity based on other conditions/examples. E.g. use random forests.

7. **Study of linear regression using median square error (as opposed to mean square error), evolutionary approaches**

In general, fitting of linear regression lines with some optimisation method (differential evolution, genetic algorithms, etc)
8. **Perform a computational experimentation on different interestingness measures for association rules, looking at their pairwise correlation, as well as stability under small changes in the data.**

Study the different measures in different conditions. Attempt to find some robust measures. Stress on visualisation techniques to cover the parameter spaces.

9. **Design and implement a system for automatic descriptive characterisation of data:**

   Develop a system for automatic data characterisations. Recommendation – try to create a UNIX style library of independent tools for each task. I.e. that each tool does one concrete thing well. These can then be used to glue together by scripting in different combinations. I/O does not have to be through pipes, this is merely a generic design principle 😊

   Overall product should be designed so that it can take in input data in some specific formats, and runs suitable analyse/scripts and decides for which kinds of summaries to build. After deciding which summary is the best, it builds automatic characterisations of the data. E.g. what type is data; how many data points, different values, ranges, 5-point summaries, histograms, density plots, scatter plots, line graphs, etc.

   Input format: Single individual “columns” in text file; TAB-separated text file

   You are allowed and welcome to use R and other existing tools. But the output should be fully automatic and correct in the sense that it should correctly call those tools and provide a report summarizing data. Where possible, create graphical summaries of data and a final PDF report

   Apply this tool for 4-5 different data sets from public data mining data set repositories.

   Team size: 3-5 people; but everyone should claim authorship for a major part of the project.

10. **Automatic concept hierarchies**

    Almost as previous, but focus is on generating automatic concept hierarchies for different data types.

    Based on those generate star/snowflake schemas automatically.

11. **Automatic analysis of US census data**

    Analyse US census data sets, and find automatically strong correlations and interesting features in data relative to income (sex, education, age, ...). Is higher education worthwhile? From which age? Etc...
12. Scalable K-medoids using triangle inequality

Implement a scalable k-medoids algorithm by trying to apply indexing of intermediate results and applying triangle inequality of metric spaces to estimate the distances without calculating them. Can this be turned into density-based approach? Provide visualisation aids.

13. Study and visualiser of different clustering methods for density-based clustering

Generate different types of data sets and test density based clustering methods. Visualise (e.g. get inspiration from ML Gallery).

14. Density based clustering

Design and implement density based clustering method(s).

To demonstrate their use, generate synthetic data as follows: draw some complex contours on 2-D, and synthesize data varying levels of density in the area vs. background. E.g. generate uniformly points, but discard them with predefined probability when outside the region.

Make an effort to visualise data (e.g. scatter plot!) and your clustering results.

Can you improve your method by giving these density differences as parameters to clustering algorithm?

Can you detect overlapping clusters (e.g. overlay two datasets)?

How does your method scale to 3-D?

15. Test and improve on DBSCAN. Consider examples where DBSCAN does not perform well out of the box. Propose some ideas about how to possibly modify the method. Demonstrate results.

Consider seriation - define 2-3 objective measures and optimise matrix reordering for these measures (e.g. using exhaustive search, genetic algorithms or differential evolution)

16. Extend K-Means into a density based clustering

Using mldeemos software for visualising machine learning and clustering methods it seemed that k-means could be extended to density based clustering.

- cluster data into (very) high number of k-means clusters. E.g. 1000 or n/5 (to be validated).
- Devise a strategy that asks if two k-means clusters are “next to each other” and whether they should be merged.
- Perform such pairwise merging of individual clusters.
• Visualise (static, somewhat like mldemos)

17. **Density based clustering on different densities.**

Think about the situation when DBSCAN does not perform well out of the box. Propose some ideas about how to possibly modify the method (this can later well turn into a project testing DBSCAN and some alternative variations of it).

18. **SPEXS – finding frequent patterns in text**

(enumerative search for frequent sequence patterns)

Implement a version of a SPEXS algorithm (originally developed by Jaak Vilo) using top-down implementation, i.e. implement the high-level abstract code and add functionality as needed. Note that depending on parameter settings, different extensions of patterns are allowed. Be modular in design. Preferably use C for efficiency.

Supervisor: Jaak Vilo

19. **ML Gallery builder**


Develop your own solution for creating similar gallery/galleries on different training data sets.

Include data generator that generates data according to some distributions and filters out training and test data sets.
Use WEKA command-line

20. **Random Forest – good or bad?**

Describe the Random Forest procedure. Identify its strengths and weaknesses and construct respective test cases to demonstrate that. Ideally find cases/examples where RF consistently outperforms many other ML methods, as well as counterexamples, where RF consistently underperforms some other ML algorithm.

Come up with a nice educational visualisation of a decision tree, multiple trees, and a random forest classifier (to supersede ML Gallery)

21. **Self Organising Maps**

Implement Self Organising Maps and perform visualisation for educational purposes (1-D, 2-D, 3D). Study the behaviour and make some educational slideshows and on-line demos.

22. **Projects with commercial companies**

I have some contacts with commercial companies. E.g. EMT, Elisa, SwedBank, etc.

A good team may be put in contact and under confidentiality agreements asked to collaborate with the company.