Project EDAMI

Contact information:

Robert Bembenik, PhD, room 301, tel. 7715, email: robert.bembenik@pw.edu.pl

Consultations: preferred online teams or zoom meetings during project consultation hours (Tuesdays, 12:15-14:00). To schedule a meeting please contact me via teams or email in advance.

The aim of the project

The aim of the project is to prepare, **implement** and verify experimentally one of data mining algorithms (e.g. clustering, classification, frequent patterns discovery, association rules discovery, etc.) presented during the lectures. Depending on the topic, the project may be realized by one or two students.

Please note!

Before the implementation you need to find the paper which you will be implementing (e.g. using google scholar).

The implementation of an algorithm refers to the process of translating the algorithm's theoretical design (as described in scientific papers referenced in each proposed topic) into a functional program or code that a computer can execute. It involves writing the specific steps of the algorithm in a programming language, ensuring that it performs the intended tasks efficiently and accurately.

Key Steps in Implementing an Algorithm

1. **Understanding the Algorithm**: Before coding, it's crucial to fully understand the logic and steps of the algorithm, its input and output requirements, and its computational complexity.
2. **Choosing a Programming Language**: Select a suitable programming language based on the needs of the application, performance requirements, and the developer's familiarity. Common languages for algorithm implementation include Python, Java, C++, and others.
3. **Writing the Code**: Translate the algorithm into code by breaking down its steps into functions, loops, conditionals, and other structures that the chosen programming language supports.
4. **Testing and Debugging**: After coding, test the algorithm with various inputs to ensure it works correctly and efficiently. Debugging involves fixing any errors or unexpected behaviors in the code.

Using existing libraries (e.g. scikit-learn) is not the implementation!

Important dates

26.11.2024 – deadline for choosing a project topic (by email, please include [EDAMI project] in the title of the message). Please include three preferred topics in the message in the order of your preference. The projects will be assigned according to the time I receive the reservation and the preference. For projects realized by teams please address the message to all members of the team.

17.12.2024 – submission of description of proposed solution, selected algorithms (in pseudocode), proposed implementation technology and experiments (Part 1). The documentation for Part 1 should also be presented during consultations by this date.

14.01.2025 – the date for submitting the final version of the project and documentation (Part 2). The discussion of the achieved results should take place during *consultations* by this date**.**

Project grading rules

The maximal number of points from Part 1 is **15** and from Part 2 is **30** (overall number of points from both parts: **45**). The final grade depends on the total number of obtained points, according to the ranges given below:

- 0 – 20: (ndst) **2.0**
- 21 – 25: (dst) **3.0**
- 26 – 30: (+dst) **3.5**
- 31 – 35: (db) **4.0**
- 36 – 40: (+db) **4.5**
- 41 – 45: (bdb) **5.0**

Documentation

Documentation should include the following sections:

1. Problem definition and introduction
2. Description of the proposed algorithm/solution with reference to the literature
3. Description of the implementation
4. User's manual (how to start/run the project)
5. Description of the used datasets
6. Experimental results (presenting properties of the proposed solution)
7. Conclusions
8. Bibliography

Experimental results should be performed using publicly available datasets, e.g.:

- https://archive.ics.uci.edu/ml/datasets.php,
- http://fimi.uantwerpen.be/data/,
- https://github.com/deric/clustering-benchmark,
- http://www.philippe-fournier-viger.com/spmf/index.php?link=datasets.php

Implementation of the algorithms/solutions should be prepared in one of the popular programming languages. Both parts of the documentation should be uploaded using dedicated slots in the leon course https://leon.pw.edu.pl/course/view.php?id=1643 .

Proposed topics

1. **Implementation of the Apriori algorithm for discovering strong association rules**
   Description: The algorithm should be implemented according to its definition provided in [1].
   Proposed number of students: 1.

2. **Implementation of one of the algorithms for associative classification (that is, classification using associative rules)**
   Description: The task is to implement CMAR algorithm for discovering associative classification rules using approach proposed in [4]. The project can include also a comparative experiments on other algorithms than CMAR for discovering associative classification rules. In such a case the project may be realized by two students.
   Proposed number of students: 1 (or 2).

3. **Discovering frequent itemsets and closed frequent itemsets**
   Description: The task is to implement algorithms discovering frequent itemsets and closed frequent itemsets according to [6].
   Proposed number of students: 2.

4. **Generating non-redundant association rules based on closed frequent itemsets.** The proposition of the possible approach is given in [7].
   Proposed number of students: 2.

5. **Implementation of DBSCAN algorithm for data clustering**
   Description: DBSCAN algorithm should be implemented according to [2].
   Proposed number of students: 1

6. **Implementation of TI-DBSCAN algorithm for more efficient data clustering**
   Description: the project may be combined with project no. 4. The TI-DBSCAN algorithm should be implemented according to [3].
   Proposed number of students: 1

7. **Discovering clusters with neighbour-based clustering and triangle inequality**
   Description: In [5] an efficient algorithm for neighbour-based clustering has been proposed. The task is to implement the algorithm proposed in [5].
   Proposed number of students: 1

8. **The efficient implementation of OPTICS algorithm**
   The OPTICS algorithm should be implemented according to its definition in [8].
   Proposed number of students: 1

9. **Partitioning data with k-means algorithm**
   The task of the project is to prepare implementation of k-means algorithm for partitioning data. The description of k-means algorithm has been given [9].
   Proposed number of students: 1

10. **Neighborhood-based clustering**
    The task is to implement the algorithm presented in [12].
    Proposed number of students: 1

11. **Discovering significant sequential patterns with SPADE algorithm**
    The SPADE algorithm should be implemented according to its definition in [10].
    Proposed number of students: 1

12. **Implementation of a Bayesian classifier**
    The task is to implement a Bayesian classifier as described in [9].
    Proposed number of students: 1

13. **Comparative analysis of several classification algorithms**

The aim of the project is to compare results of three classification approaches: Naïve Bayesian Classifier, Lazy Classification with Contrast Patterns (eventually with SPRINT algorithm for building Decision Trees) [9, 11].
Proposed number of students: 2 (maximum 3).

14. Topic proposed by the student and agreed upon with the instructor

References

1. Rakesh Agrawal and Ramakrishnan Srikant. 1994. Fast Algorithms for Mining Association Rules in Large Databases. In Proceedings of the 20th International Conference on Very Large Data Bases (VLDB '94).
2. Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD'96).
3. Kryszkiewicz M., Lasek P. (2010) TI-DBSCAN: Clustering with DBSCAN by Means of the Triangle Inequality. In: Szczuka M., Kryszkiewicz M., Ramanna S., Jensen R., Hu Q. (eds) Rough Sets and Current Trends in Computing. RSCTC 2010.
4. Wenmin Li, Jiawei Han and Jian Pei, "CMAR: accurate and efficient classification based on multiple class-association rules," Proceedings 2001 IEEE International Conference on Data Mining, San Jose, CA, USA, 2001, pp. 369-376.
5. Kryszkiewicz M., Lasek P. (2010) A Neighborhood-Based Clustering by Means of the Triangle Inequality. In: Fyfe C., Tino P., Charles D., Garcia-Osorio C., Yin H. (eds) Intelligent Data Engineering and Automated Learning – IDEAL 2010. IDEAL 2010
6. Mohammed J. Zaki and Ching-Jui Hsiao. 2005. Efficient Algorithms for Mining Closed Itemsets and Their Lattice Structure. IEEE Trans. on Knowl. and Data Eng. 17, 4 (April 2005), 462-478
7. Mohammed J. Zaki. 2000. Generating non-redundant association rules. In Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '00). ACM, New York, NY, USA, 34-43
8. Mihael Ankerst, Markus M. Breunig, Hans-Peter Kriegel, and Jörg Sander. 1999. OPTICS: ordering points to identify the clustering structure. In Proceedings of the 1999 ACM SIGMOD international conference on Management of data (SIGMOD '99)
9. Jiawei Han, Micheline Kamber, and Jian Pei. 2011. Data Mining: Concepts and Techniques (3rd ed.). Morgan Kaufmann Publishers Inc., San Francisco, CA, USA
10. Zaki, M.J. Machine Learning (2001) 42: 31
11. John C. Shafer, Rakesh Agrawal, and Manish Mehta. 1996. SPRINT: A Scalable Parallel Classifier for Data Mining. In Proceedings of the 22th International Conference on Very Large Data Bases (VLDB '96)
12. Shuigeng Zhou, Yue Zhao, Jihong Guan, Joshua Zhexue Huang: A Neighborhood-Based Clustering Algorithm. PAKDD 2005: 361-371