ArmSquare: An Association Rule Miner Based on Multidimensional Numbered Information Spaces

Iliya Mitov[1], Krassimira Ivanova[1], Benoit Depaire[2], Koen Vanhoof[2]

1: Institute of Mathematics and Informatics – BAS, Sofia, Bulgaria

2: Hasselt University, Belgium

Introduction

Data mining stands at the crossroad of databases, artificial intelligence, and machine learning.

Association rule mining (ARM) is a popular and well researched method for discovering interesting rules from large collections of data.

Applied in - market basket analysis, gene-expression data analysis, building statistical thesaurus from the text databases, finding web access patterns from web log files, discovering associated images from huge sized image databases, etc.

The efficiency of frequent itemset mining algorithms is determined mainly by three factors:

(1) the way candidates are generated;

(2) the data structure that is used;

(3) the implementation details.

** The work was supported in part by Hasselt University under the Project R-1876 and by Bulgarian NSF under the project D002-308.*

Previous Works

- Apriori [Agrawal and Srikant, 1994]: the best-known ARM. Uses a breadth-first search strategy to count the support of itemsets.
- AGM [Inokuchi et al, 2003]: finds all frequent induced sub-graphs with a vertex-growth strategy.
- FSG [Kuramochi and Karypis, 2001]: uses edge-growth strategy.
- gSpan [Yan and Han, 2002]: uses a depth-first search for finding candidate frequent sub-graphs.
- ECLAT [Zaki et al, 1997]: uses a depth-first search; the first algorithm that uses a vertical data (inverted) layout.
- FP-Tree [Han and Pei, 2000]: extended prefix-tree structure storing quantitative information about frequent patterns.
- TreeProjection [Agarwal et al, 2000]: uses lexicographical tree.
- Hash-based techniques for candidate generation – DHP (Direct Hashing and Pruning) [Zaki et al, 1997]; PHP (Perfect Hashing) [Özel and Güvenir, 2001]; IHP (Inverted Hashing and Pruning) [Holt and Chung, 2002].
- Hmine [Pei et al, 2001]: introduces the concept of hyperlinked data structure (array-based structure).

Pros-cons of Different Structures

- Tree-based structures:
  - \+ reduce traversal cost
  - \- incur high construction cost, especially in sparse large datasets
- Array-based structures:
  - \+ incur little construction cost
  - \- need much more traversal cost

ArmSquare

- The approach is focused on proposing appropriate coding of the items in database in order to use the possibilities of direct access to the information via coordinate vectors into multidimensional numbered information spaces.
- These structures combines the convenience of the work with array structures with economy of tree structures that lies in realization of the access method.
- The algorithm of obtaining association rules is very simple; we focus our attention over the possibilities of using such structures for storing information in data mining systems.

ArmSquare:

- ARM - used in literature for short denotation of "association rule miner"
- ArM ("Archive Manager) - access method that realizes Multidimensional Numbered Information Spaces

Multidimensional Numbered Information Spaces

- Constructs:
  - basic information *elements*
    (an arbitrary long sequence of machine codes)
  - numbered information *spaces*
    (organization of basic information elements in array-like structures, but on hierarchical realization)

    Each element is accessed by coordinate array $A=(n,p_1,\ldots,p_n)$
    n is the dimension (variable) and $p_1,\ldots,p_n$ are coordinates.
  - *indexes* and *meta-indexes* (sequences of space addresses)
  - Special kind of space index is a *projection* (analytical given index)
    - *hierarchical projection* – (fixed the top part of coordinates)
    - *arbitrary projection* (fixed coordinates in arbitrary positions).

Multidimensional Numbered Information Spaces

- Operations:
  - *with elements* - updating (all elements virtually exist), getting the value, getting length, and positioning in the element;
  - *with spaces* - copying the first space in the second and moving the first space in the second with modifications specifying clearing or remaining the second space before operation
  - using the *hierarchical projection* - crawling the defined area and extracting next/previous empty/non-empty elements; receiving the whole index or its length, of the non-empty elements.
  - for *arbitrary projection* - the same operations (only for non-empty elements)

ArmSquare - **pretreatment**; data processing; analysis and monitoring

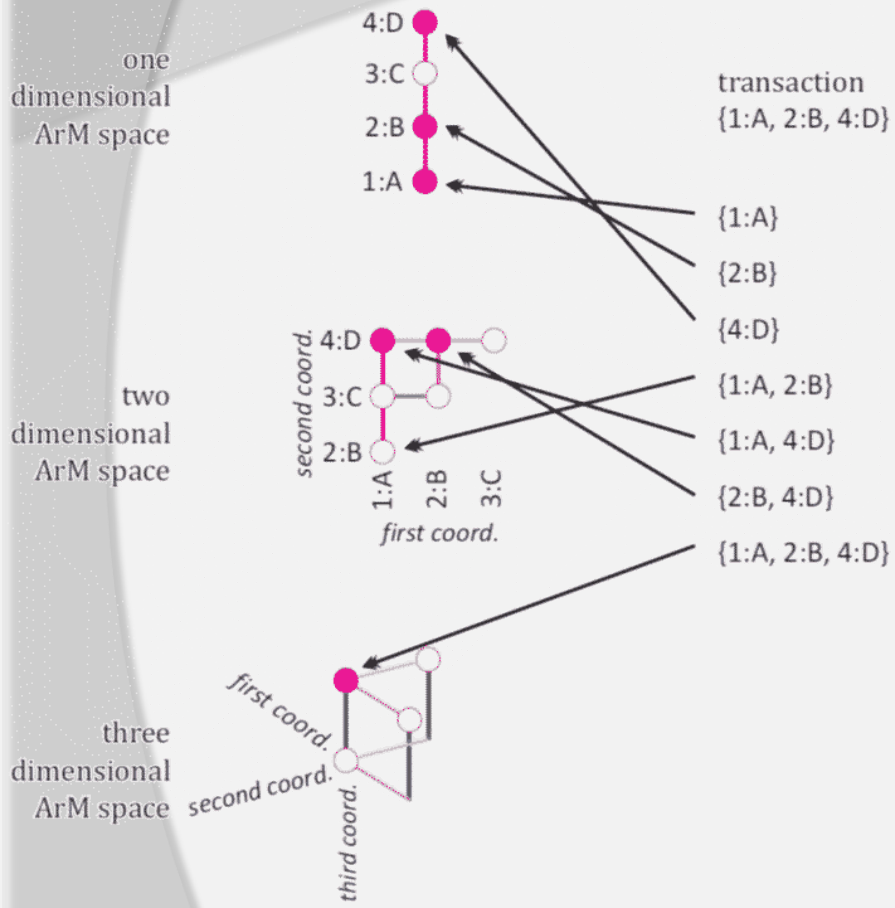
- Creating a mapping between incoming items and natural numbers by order of fist occurrence of the item.
- Each incoming transaction is "coded" by these numbers.
- The items in each transaction are sorted in increasing order.

{A,B}, {C,D}, {D,B,A}, {B,C}, {C,D,B}

{1,2}, {3,4}, {4,2,1}, {2,3}, {3,4,2}

{1,2}, {3,4}, {1,2,4}, {2,3}, {2,3,4}

ArmSquare -
pretreatment; **data processing**; analysis and monitoring



Result of data processing of the database

transactional dataset

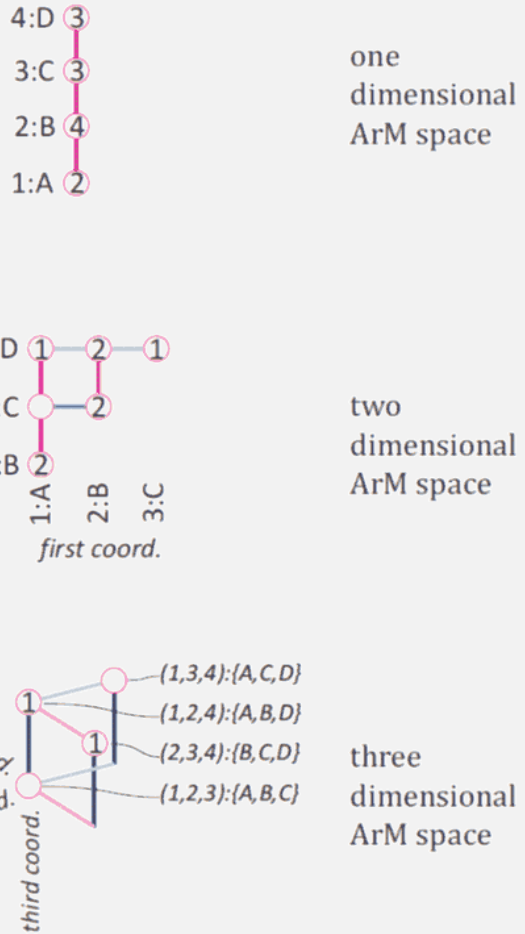
{1:A, 2:B}

{3:C, 4:D}

{1:A, 2:B, 4:D}

{2:B, 3:C}

{2:B, 3:C, 4:D}



Accumulating in ArM spaces
of the number of occurrence
of produced itemsets from one transaction

ArmSquare -
pretreatment; data processing; **analysis and monitoring**

- For obtaining all existing k-itemset with a support of at least *MinSup*, crawling over the k-dimensional ArM-space is done using the function *ArmNextProj*, starting with hierarchical projection (-,-,...,-).
- Using the function *ArmRead* for the current extracted non-empty element, the value is read and is compared with *MinSup*.
- If this value is no less than *MinSup*, the corresponded itemset is included into the resulting list of itemsets.
- The resulting itemsets is sorted by decreasing support.

Similarly, extracting the rules with more than *MinConf* (using values of addresses of heads and bodies, generated from intemsets) are made.

Advanced Specifics of ArmSquare

- In Apriori algorithm min-support is set. In a higher value of min-support Apriori is highly convergent, while a small amount of min-support leads to almost total exhaustion of short itemsets.
- In ArmSquare, after building the spaces, statistics for min-support for each area can be derived separately, which allows to give for further analysis different min-support for different numbers of elements in combinations.
- Usually, ARM-approaches derive all successive combinations in ascending order and changing the min-support causes a repetition of the whole algorithm.
- In ArmSquare, structuring the itemsets support in ArM-space allows subsequent analysis to be made very quickly by setting a different min-support and profiles of different lengths of itemsets.
- The information for itemsets with particular length containing a specific element can be directly extracted.
- The database can be interactively expanded as well as the processing of the transactions can be made in parallel.

Applications

- Retail market basket data set supplied by anonymous Bulgarian retail supermarket store.
  - one year period (2008 year); middle supermarket; town with about 30 000 citizens; total number of transactions - 108 846; number of items - 3 609; maximum transactional length - 23.
  - grouped in accordance of months - analyzing the deviations of purchasing during the months.
- Dataset that included several types of color harmonies and contrast features, extracted by 600 paintings of 19 artists from different movements of West-European fine arts and Eastern Medieval Culture.
  - using the possibility of binning the dataset by class label allowed to use ArmSquare as element in the generation rule phase of CAR-algorithm and extract typical combinations of features for examined artists.

Conclusion

- The main focus in the realization of ArmSquare is to show the possibilities to use the advantages of multi-dimensional information spaces for memory structuring in the area of data mining and knowledge discovery.
- The variety of the tasks that can be made with proposed frequent association rule miner ArmSquare allows comprehensive and facile analysis of the situations and conducting forecasting in wide areas of applications.
- The next steps will be focused on improving the algorithm of extracting rules, especially realizing the ARUBAS algorithm [Depaire et al, 2008] over the multi-dimensional information spaces.

Bibliography

- Bodon, F., "A fast APRIORI implementation". In IEEE ICDM Workshop on FIMI, Melbourne, Florida, USA, 2003.
- Agrawal, R., Imieliński, T., and Swami, A., "Mining association rules between sets of items in large databases". In Proc. of the ACM SIGMOD ICMD, Washington, DC, 1993, pp. 207-216.
- Goethals, B., Efficient Frequent Pattern Mining. PhD thesis in Transnationale Universiteit Limburg, 2002.
- Agrawal, R. and Srikant, R., "Fast algorithms for mining association rules", In Proc. of the 20th Int. Conf. on VLDB, 1994, pp. 487-499.
- Inokuchi, A., Washio, T., and Motoda, H., "Complete mining of frequent patterns from graphs: mining graph data". In Machine Learning, Vol.50, 2003, pp. 321-354.
- Kuramochi, M. and Karypis, G., "Frequent subgraph discovery". In Proc. of the 1st IEEE Int. Conf. on DM, 2001, pp. 313-320.
- Yan, X. and Han, J., "gSpan: Graph-based structure pattern mining". In Proc. of the 2nd IEEE Int. Conf. on DM, 2002, pp. 721-724.
- Zaki, M., Parthasarathy, S., Ogihara, M., and Li, W., "New algorithms for fast discovery of assoctiation rules". In Proc. of the 3rd Int. Conf. on KD and DM, 1997, pp. 283-286.
- Park, J., Chen, M., and Yu, P., "An effective hash based algorithm for mining association rules". In Proc. of ACM SIGMOD Int. Conf. on Management of Data, 24/2, 1995, pp. 175-186.
- Özel, S. and Güvenir, H., "An algorithm for mining association rules using perfect hashing and database pruning". In Proc. of the TAINN, 2001, pp. 257-264.
- Holt, J. and Chung, S., "Mining association rules using inverted hashing and pruning". Information Processing Letters Archive, 83/4, 2002, pp. 211-220.
- Han, J. and Pei, J., "Mining frequent patterns by pattern-growth: methodology and implications. In ACM SIGKDD Explorations Newsletter 2/2, 2000, pp. 14-20.
- Pei, J., Han, J., Lu, H., Nishio, S., Tang, S., and Yang, D., "Hmine: hyper-structure mining of frequent patterns in large databases". In Proc. of IEEE ICDM, 2001, pp. 441-448.
- Agarwal, R., Aggarwal, C., and Prasad V., "A tree projection algorithm for generation of frequent item-sets". In Journal of Parallel and Distributed Computing, 61/3, 2000, pp. 350-371.
- Bodon, F. and Ronyai, L., "Trie: an alternative data structure for data mining algorithms". In Mathematical and Computer Modelling, 38/7, 2003, pp. 739-751.
- Yuan, Y. and Huang, T., "A Matrix algorithm for mining association rules". In LNCS, Vol. 3644, 2005, pp. 370-379.
- Liu, G., Lu, H., Yu, J., Wang, W., and Xiao, X., "AFOPT: An efficient implementation of pattern growth approach". In Workshop on Frequent Itemset Mining Implementation. (FIMI 03), 2003.
- Mitov, I., Ivanova, K., Markov, K., Velychko, V., Vanhoof, K., and Stanchev, P., "PaGaNe – a classification machine learning system based on the multidimensional numbered information spaces". In WSPS on CEIS, No. 2, 2009, pp. 279-286.
- Markov, K., "Multi-domain information model". In Int. J. on Information Theories and Applications, 11/4, 2004, pp. 303-308.
- Markov K., Ivanova, K., Mitov, I., and Karastanev, S., "Advance of the access methods". Int. J. on Information Technologies and Knowledge, 2/2, 2008, pp. 123-135.
- Ivanova K., Stanchev P., and Vanhoof K., "Automatic tagging of art images with color harmonies and contrasts characteristics in art image collections". Int. J. on Advances in Software, 3/3&4, 2010, pp. 474-484.
- Depaire, B., Vanhoof, K., and Wets, G., "ARUBAS: an association rule based similarity framework for associative classifiers". In IEEE Int. Conf. on Data Mining Workshops, 2008, pp. 692-699.

Thank you for the attention!

Iliya Mitov, Krassimira Ivanova, Benoit Depaire, Koen Vanhoof

ArmSquare: An Association Rule Miner
Based on Multidimensional Numbered Information Spaces