University of Maribor

Faculty of Electrical Engineering
and Computer Science

FERI

Data squashing as preprocessing in association rule mining

IEEE Symposium Series On Computational Intelligence
(SSCI 2022)
Nature-Inspired Computation In Engineering (IEEE NICE)

Authors:
Iztok Fister (iztok.fister@um.si), Iztok Fister Jr. (iztok.fister1@um.si), Damijan Novak (damijan.novak@um.si) and Domen Verber (domen.verber@um.si)

7.12.2022

PRESENTATION AGENDA

1. Motivation and purpose of the article
2. Data squashing
3. Association Rule Mining (ARM)
4. ARM data squashing:
   - ❑ similarity metrics,
   - ❑ design of the ARM data squashing algorithm,
   - ❑ implementation of the ARM data squashing algorithm.
5. Experiments, results and discussion
6. Future work

MOTIVATION

- Large databases are available from social media platforms, IoT sensors, public repositories, cloud storage, logging files, traditional databases, and many others.

- It is resource (time, memory space, and processing power) intensive to process such databases.

- Methods are needed, which:
  - ❑ reduce the sizes of such databases,
  - ❑ require fewer data analysis resources, and
  - ❑ produce the same results as the original.

PURPOSE

- The new preprocessing method for data squashing is proposed.

- Method is based on the Cosine and Euclidean distance similarities.

- Integration of the method to the uARMSolver framework, and therefore widening the usability of it to a broad spectrum of users.

DATA SQUASHING

- Defined by DuMouchel et al., data squashing is the construction of a smaller dataset providing approximately the same data analysis results as the large dataset.

W. DuMouchel, C. Volinsky, T. Johnson, C. Cortes, and D. Pregibon,

“Squashing flat files flatter,” in KDD ’99, 1999.

- The classical approach in developing these methods tries to preserve the statistical information of data in the reduced database.
- In our study, the statistical information of a data set is replaced with a concept of similarity.
- In general, there are three kinds of data squashing methods:

1. a Taylor series representation for the likelihood function, with which the arbitrary problem is modeled,
2. approximating a specific likelihood function directly by building the squashed dataset, and
3. developing the empirical likelihood method, capable of direct matching moments of the original variables in the squashed dataset to moments in the original dataset.

DATA SQUASHING (2)

- A transaction dataset Y is defined as a matrix with N rows representing transactions and K columns denoting different features of a single transaction.
- The squashed dataset is a matrix X with M rows, where $M \ll N$, and $K + 1$ columns.
- The extra column in X represents the weights w_i for $i = 1, \ldots, M$.
- The weights represent the sizes of the data clusters of similar transactions and have to satisfy the following criteria:

$$\sum_{i=1}^{M} w_i = N$$

subject to $w_i > 0$

ASSOCIATION RULE MINING (ARM)

- Association Rule Mining (ARM) is a well-known Machine Learning method. Several modern approaches to ARM are based on evolutionary and swarm intelligence algorithms.
- ARM can serve as a tool for discovering the hidden relations between attributes in a transaction database.
- Transaction = recording of some event and associated attributes.
- ARM generates the set of rules that represents the relation between the attributes in the transactions.
- Association rule is defined as an implication:

$$X \Rightarrow Y,$$

where $X \subset F$, $Y \subset F$, and $X \cap Y = \emptyset$.

$$\text{Set of features } F = \{\underset{\substack{\text{categorical}\\\text{features}}}{A1, \ldots, AC}; \underset{\substack{\text{numerical}\\\text{features}}}{Q1, \ldots, QR}\}$$

ARM DATA SQUASHING: SIMILARITY METRICS

- The idea of ARM data squashing is to collect similar transactions to the same group (cluster) based on chronological ordering.

- In our case, two measurements of the similarity between vectors x and y that form the same cluster were used:
  - The Cosine similarity is calculated according to the Cauchy-Schwartz inequality:

$$sim_{cos}(x, y) = \frac{|x \cdot y|}{\|x\| \cdot \|y\|}$$

$|x \cdot y|$ is a scalar product
$\|x\|$ and $\|y\|$ are norms of both vectors

  - The Euclidean distance similarity:

$$sim_{Eucl}(x, y) = \sqrt{\sum_{i=1}^{N} \frac{(x_i - y_i)^2}{(x_i^{max} - x_i^{min})^2}}$$

x = $\{x_i\}$ and y = $\{y_i\}$ represent items in the original transaction dataset;
$x^{\max} = max_{i=1,\cdots,N}\{x_i\}$
$x^{min} = min_{i=1,\cdots,N}\{x_i\}$ are the maximum and minimum values of the vector‘s elements;
N is the number of transactions.

ARM DATA SQUASHING: DESIGN OF THE ARM DATA SQUASHING ALGORITHM

- Integration of the ARM data squashing algorithm into a universal framework called the uARMSolver.

- Three issues were considered during integration:
  1. examining the size of the specific clusters,
  2. modifying the fitness function accordingly, and
  3. determining the representative transaction in a cluster.

- The first issue is considered by including the additional feature in the transaction dataset that identifies the weight of the cluster.

- The weight of the cluster is incorporated into the fitness function evaluation.

- When a categorical feature is observed, the frequency of appearances of an observed attribute is counted, and the most frequent attribute is taken into the representative transaction.

ARM DATA SQUASHING: IMPLEMENTATION OF THE ARM DATA SQUASHING ALGORITHM

- The transactions are processed in order from the start of the dataset to the end. The already processed values are marked as visited. In each step, the first unvisited transaction is selected. That represents the first transaction in the new cluster. After that, other unvisited transactions that satisfy the relation are added:

$$sim_{\cos|Eucl}(x, y) \leq Thresh$$

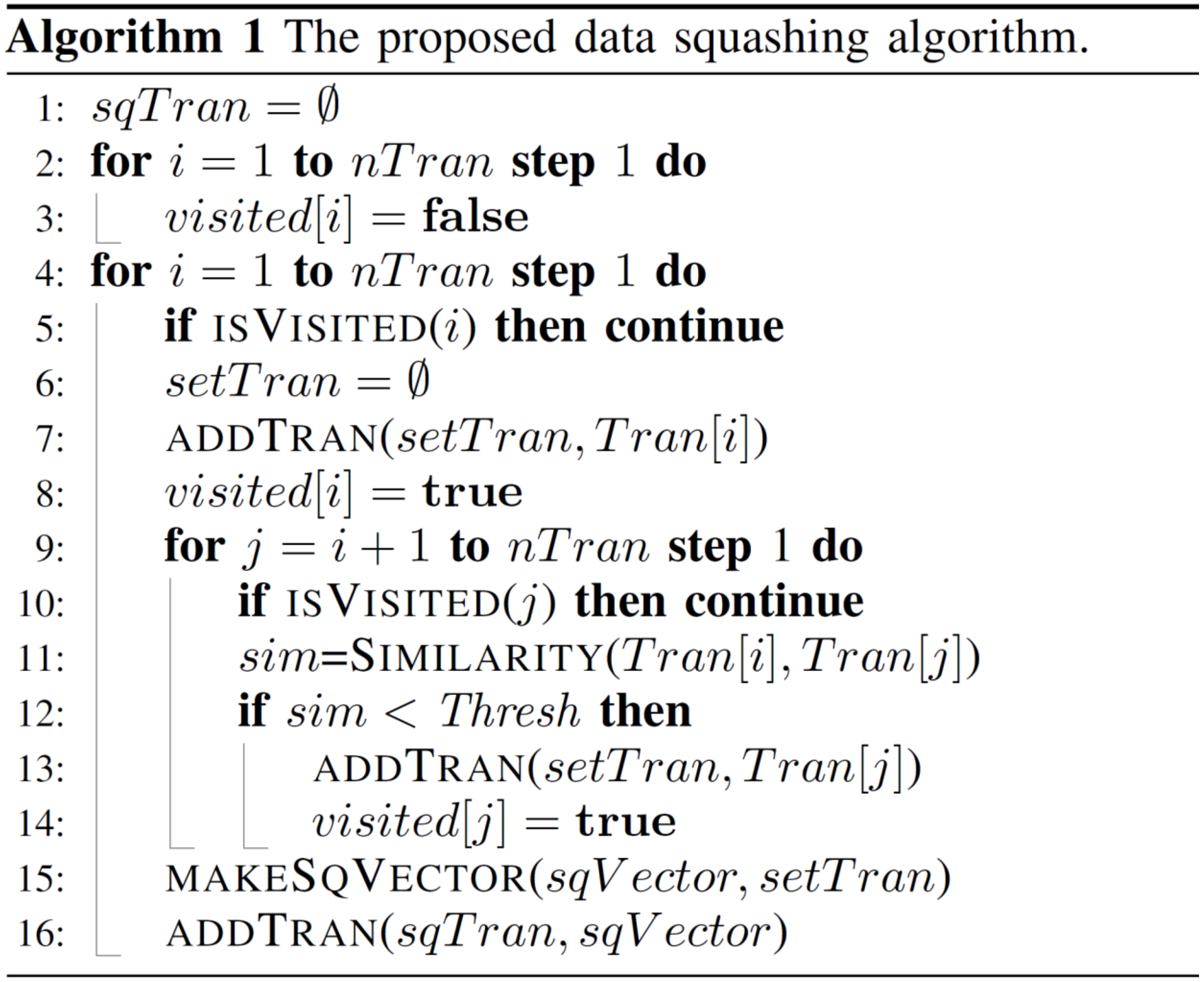
Thresh denotes the minimum similarity value allowing the specific transaction to be added to the cluster.

- The process is repeated until all unvisited transactions are checked.

- A representative transaction of the cluster is calculated.

- The representative transaction is added to the squashed dataset.

ARM DATA SQUASHING: IMPLEMENTATION OF THE ARM DATA SQUASHING ALGORITHM (2)



Algorithm 1 The proposed data squashing algorithm.

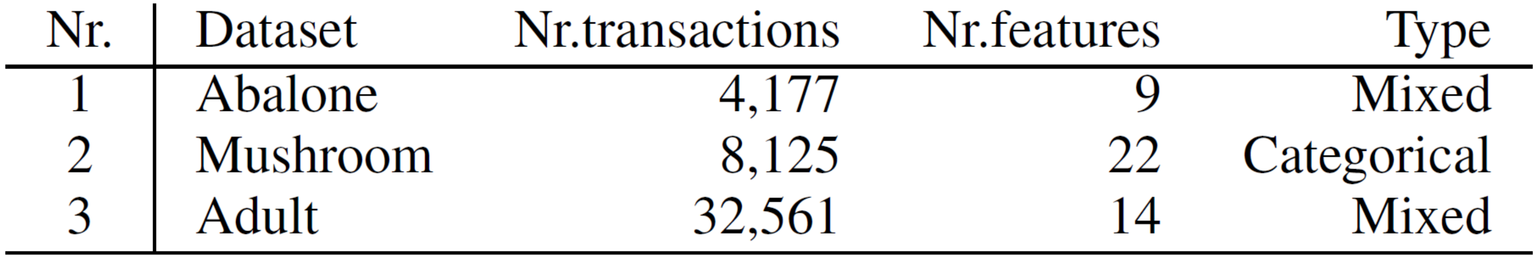
```
$sqTran = \emptyset$
for $i = 1$ to $nTran$ step $1$ do
    $visited[i]$ = false
for $i = 1$ to $nTran$ step $1$ do
    if ISVISITED($i$) then continue
    $setTran = \emptyset$
    ADDTRAN($setTran, Tran[i]$)
    $visited[i]$ = true
    for $j = i + 1$ to $nTran$ step $1$ do
        if ISVISITED($j$) then continue
        $sim$=SIMILARITY($Tran[i], Tran[j]$)
        if $sim < Thresh$ then
            ADDTRAN($setTran, Tran[j]$)
            $visited[j]$ = true
    MAKESQVECTOR($sqVector, setTran$)
    ADDTRAN($sqTran, sqVector$)
```

EXPERIMENTS, RESULTS AND DISCUSSION

- Experiments were conducted in two phases:

1. preprocessing of the data,
2. the association rules were mined with the utilization of the Differential Evolution (DE) algorithm (Np = 100, F = 0.9, CR = 0.5).

- Three UCI ML datasets were used:

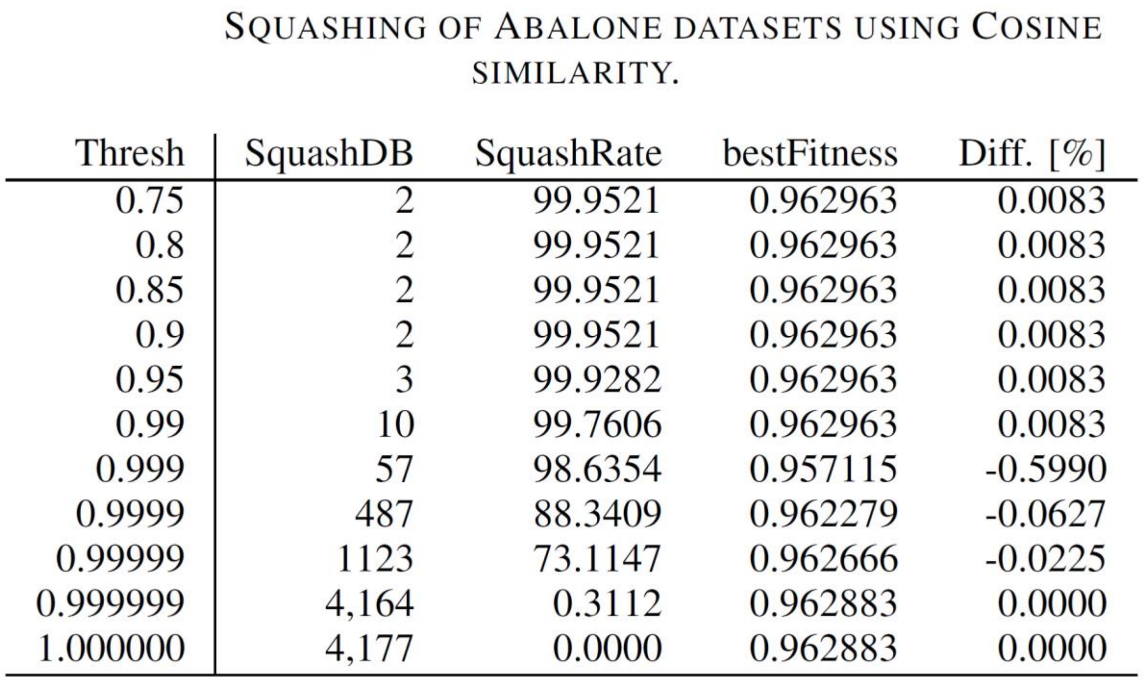


Nr.	Dataset	Nr.transactions	Nr.features	Type
1	Abalone	4,177	9	Mixed
2	Mushroom	8,125	22	Categorical
3	Adult	32,561	14	Mixed

- The results were analysed according to:

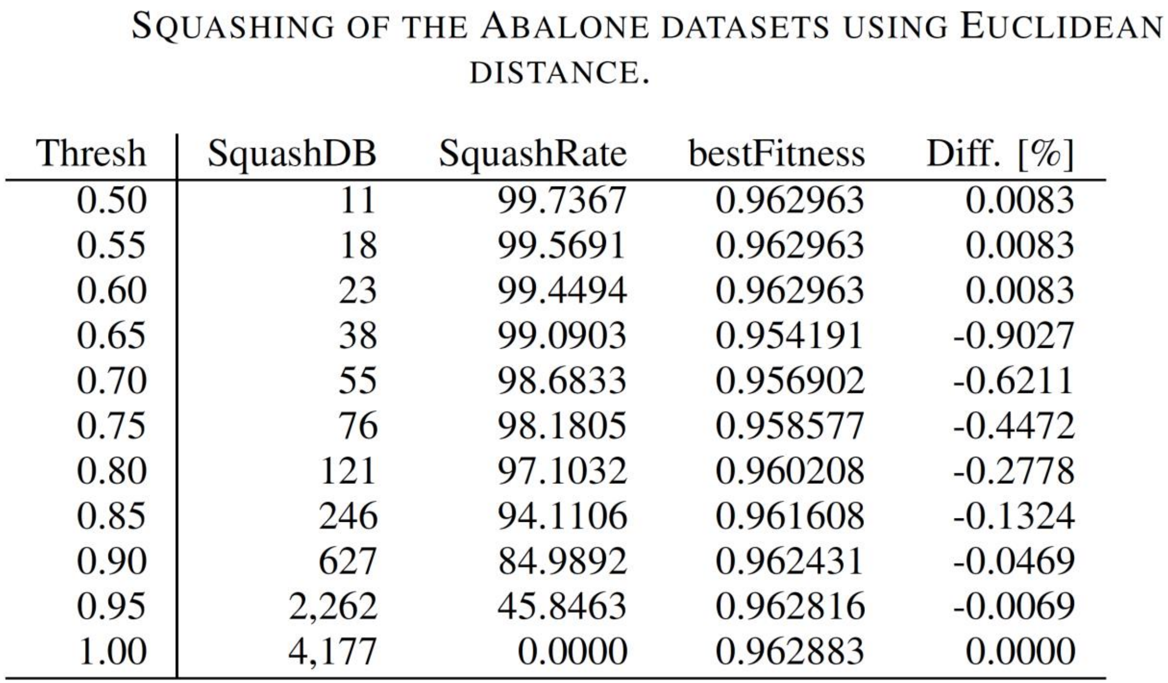
1. the quality, and
2. the time complexity.

RESULTS: REGARDING THE QUALITY



SQUASHING OF ABALONE DATASETS USING COSINE SIMILARITY.

Thresh	SquashDB	SquashRate	bestFitness	Diff. [%]
0.75	2	99.9521	0.962963	0.0083
0.8	2	99.9521	0.962963	0.0083
0.85	2	99.9521	0.962963	0.0083
0.9	2	99.9521	0.962963	0.0083
0.95	3	99.9282	0.962963	0.0083
0.99	10	99.7606	0.962963	0.0083
0.999	57	98.6354	0.957115	-0.5990
0.9999	487	88.3409	0.962279	-0.0627
0.99999	1123	73.1147	0.962666	-0.0225
0.999999	4,164	0.3112	0.962883	0.0000
1.000000	4,177	0.0000	0.962883	0.0000



SQUASHING OF THE ABALONE DATASETS USING EUCLIDEAN DISTANCE.

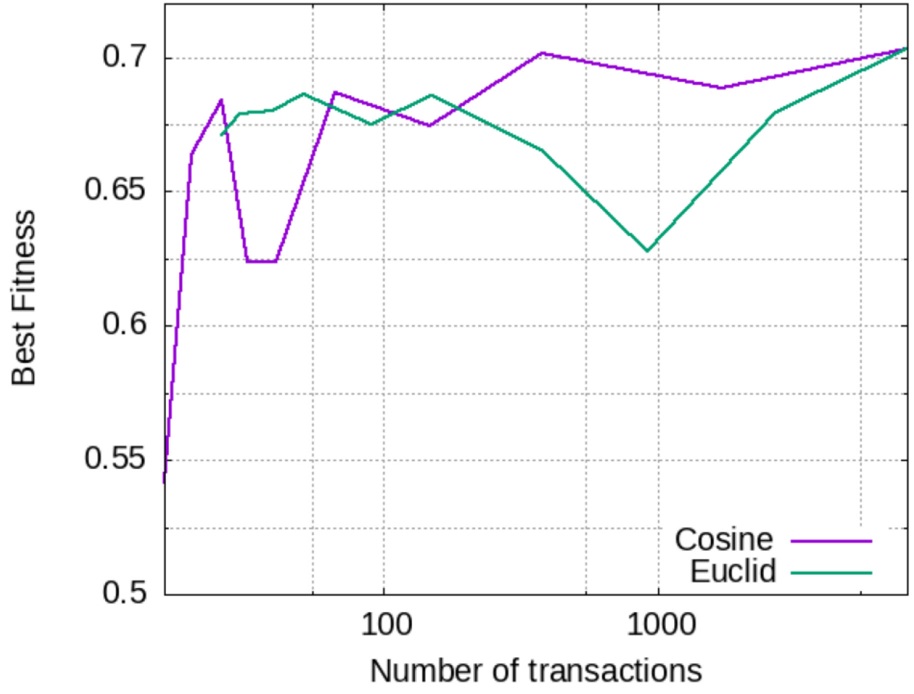
Thresh	SquashDB	SquashRate	bestFitness	Diff. [%]
0.50	11	99.7367	0.962963	0.0083
0.55	18	99.5691	0.962963	0.0083
0.60	23	99.4494	0.962963	0.0083
0.65	38	99.0903	0.954191	-0.9027
0.70	55	98.6833	0.956902	-0.6211
0.75	76	98.1805	0.958577	-0.4472
0.80	121	97.1032	0.960208	-0.2778
0.85	246	94.1106	0.961608	-0.1324
0.90	627	84.9892	0.962431	-0.0469
0.95	2,262	45.8463	0.962816	-0.0069
1.00	4,177	0.0000	0.962883	0.0000

- The influence of the *Thresh* parameter is crucial. However, the correlation between the *Thresh* parameter and the size of the squashed database is not linear for Cosine similarity.

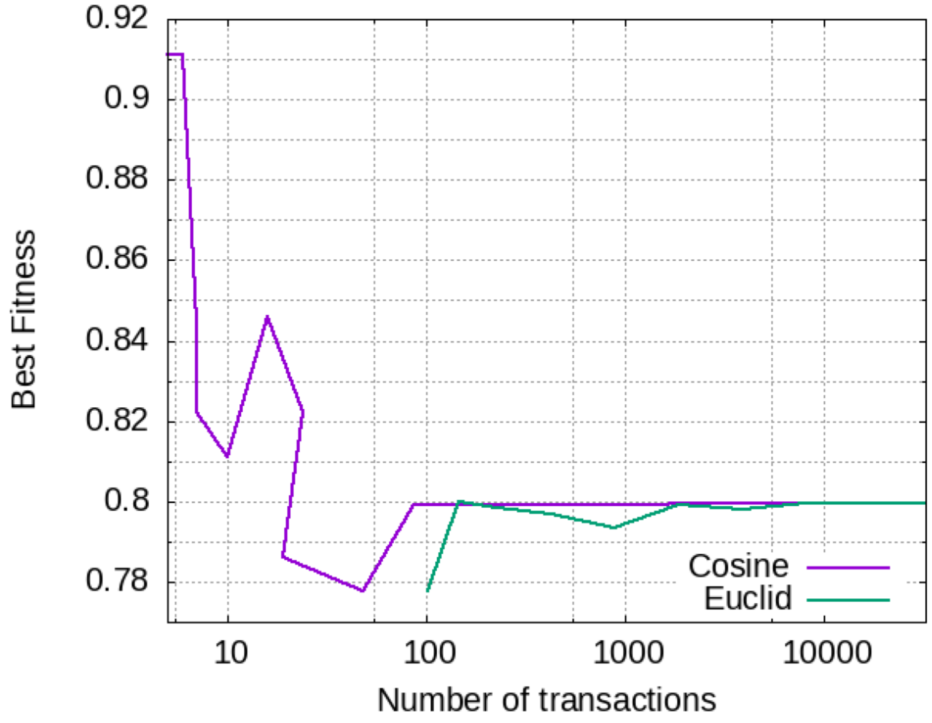
- On the other hand, some linear correlation exists for Euclidean distance.

- Some negative values of the differences can be observed in the tables. These negative values show that the ARM mining algorithm didn’t achieve the original bestFitness value in some instances of *Thresh* values. For example, the instance *Thresh* = 0.999 in the first table has performed 0.6 % less quality solution according to *bestFitness* when squashing database for more than 98 %.

RESULTS: REGARDING THE QUALITY (2)



Squashing of the Mushroom dataset.

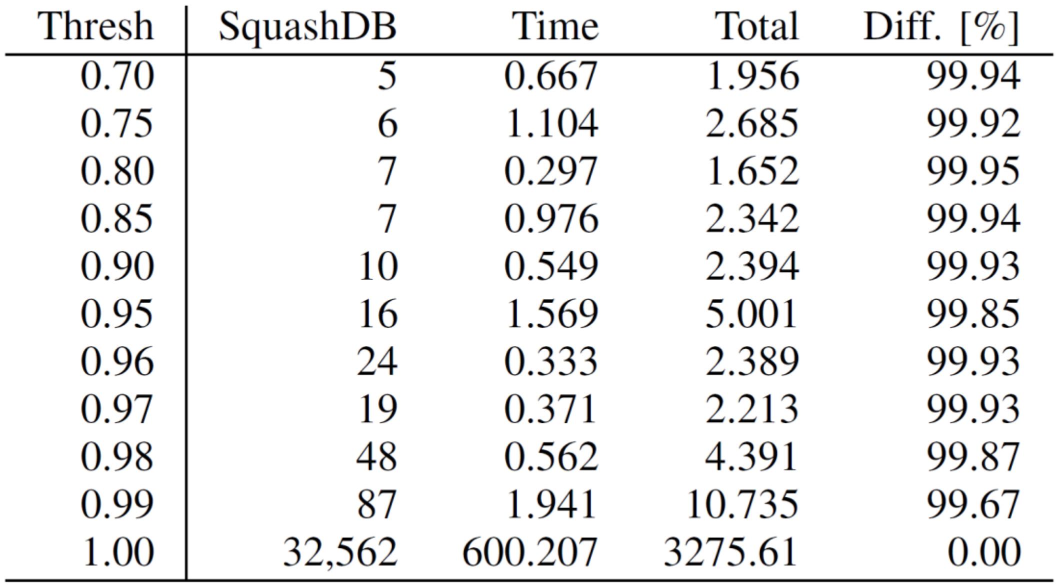


Squashing of the Adult dataset.

- The Cosine similarity metric is more sensible on the lower values of the *Thresh* parameter. On the other hand, this similarity directs the algorithm to faster convergence.
- This fact is emphasised even more in the second graph, where the best fitness function values deviate a lot around the best value by the lower settings of this parameter before convergence is achieved. The convergence is achieved immediately when the value of the *Thresh* parameter is changed from 0.99 to 1.00.

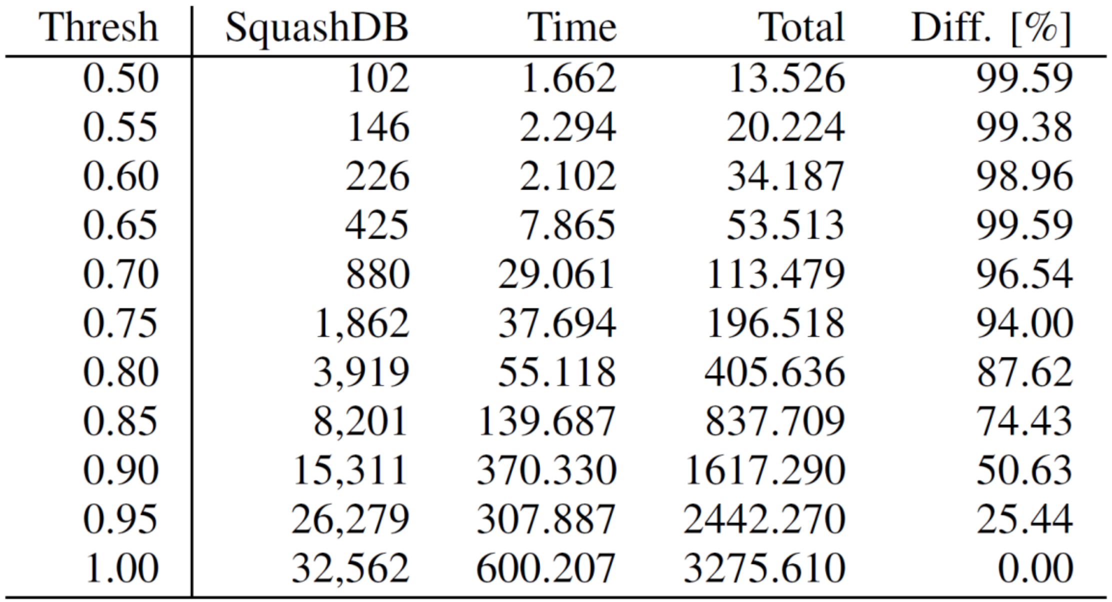
RESULTS: REGARDING THE TIME

SQUASHING OF THE ADULT DATASETS USING COSINE SIMILARITY.



Thresh	SquashDB	Time	Total	Diff. [%]
0.70	5	0.667	1.956	99.94
0.75	6	1.104	2.685	99.92
0.80	7	0.297	1.652	99.95
0.85	7	0.976	2.342	99.94
0.90	10	0.549	2.394	99.93
0.95	16	1.569	5.001	99.85
0.96	24	0.333	2.389	99.93
0.97	19	0.371	2.213	99.93
0.98	48	0.562	4.391	99.87
0.99	87	1.941	10.735	99.67
1.00	32,562	600.207	3275.61	0.00

SQUASHING OF ADULT DATASETS USING EUCLIDEAN DISTANCE.



Thresh	SquashDB	Time	Total	Diff. [%]
0.50	102	1.662	13.526	99.59
0.55	146	2.294	20.224	99.38
0.60	226	2.102	34.187	98.96
0.65	425	7.865	53.513	99.59
0.70	880	29.061	113.479	96.54
0.75	1,862	37.694	196.518	94.00
0.80	3,919	55.118	405.636	87.62
0.85	8,201	139.687	837.709	74.43
0.90	15,311	370.330	1617.290	50.63
0.95	26,279	307.887	2442.270	25.44
1.00	32,562	600.207	3275.610	0.00

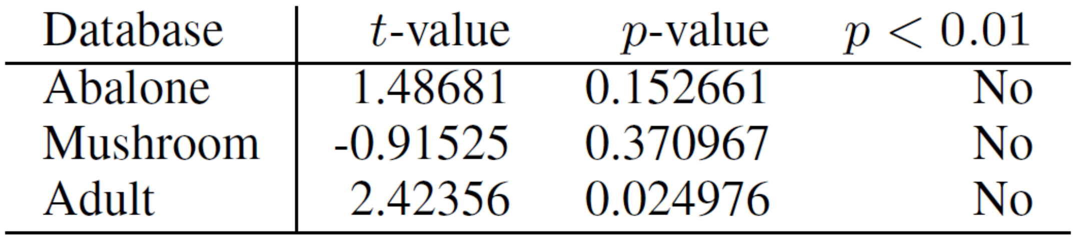
- From first table it can be seen that the execution time is decreased by data squashing by more than 99 %. However, because the dataset size is reduced significantly, it could be speculated that much information is lost.

- It can be seen from the second table that both the size of the dataset and the execution time correlates with the parameter *Thresh*.

DISCUSSION

- The proposed data squashing method based on two similarity metrics has shown their potential for application to ARM problems in practice.
- The small changes of the threshold parameter *Thresh* by the Cosine similarity can generate the huge changes in the size of the squashed dataset.
- The Euclidean distance similarity measures the distances between elements of the observed vectors in the problem search space. Consequently, this metric is more linear, scalable, and therefore easier for use in practice.
- However, the results of different metrics cannot be distinguished according to a 2-paired t-test parametric statistical test.

THE RESULTS OF THE t-TESTS.



Database	t-value	p-value	$p < 0.01$
Abalone	1.48681	0.152661	No
Mushroom	-0.91525	0.370967	No
Adult	2.42356	0.024976	No

DISCUSSION (2)

- The smaller-squashed datasets were obtained using the Cosine similarity. This is fine for the datasets with numerical attributes but can break down with the dataset containing mixed attributes.

- Cosine similarity is generally a good choice for high dimension and categorical data, while the Euclidean-based similarity measure is a good choice for numerical datasets. In addition, the Euclidean distance is not a good choice for a dataset that contains categorical attributes.

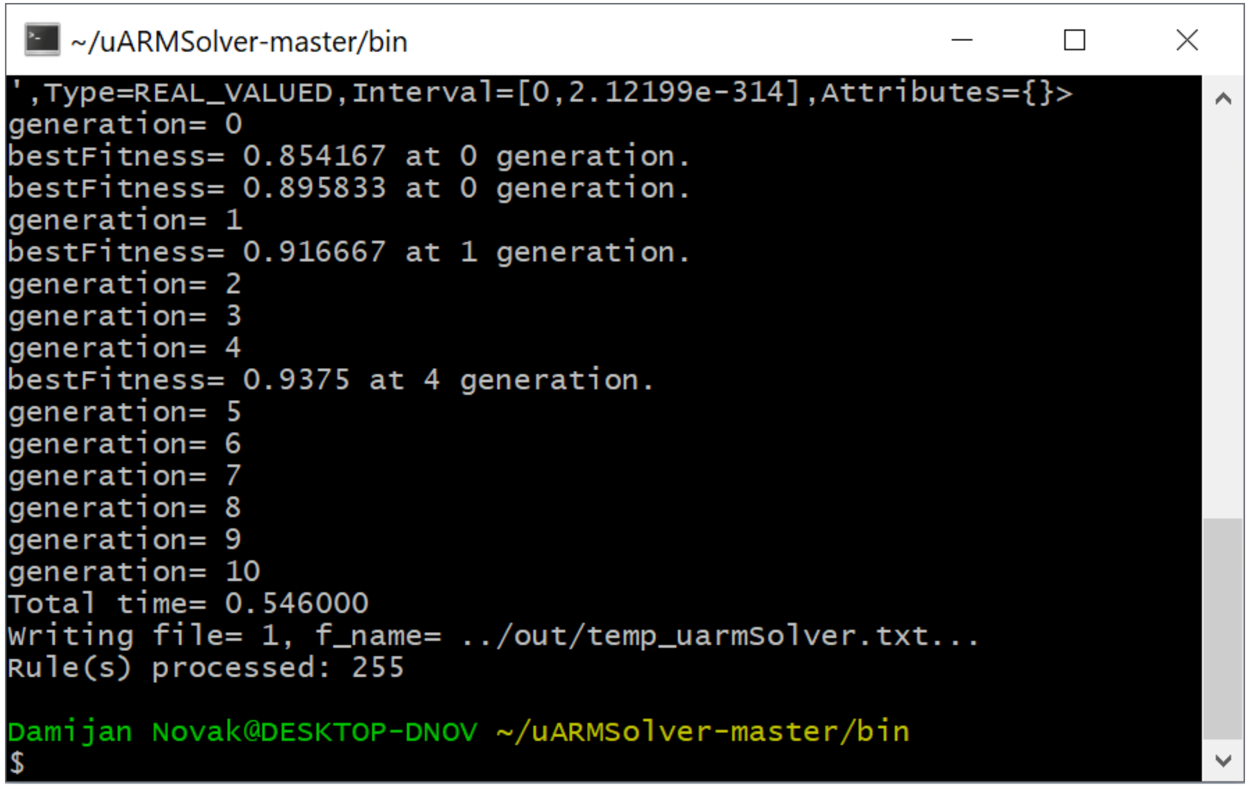
- Despite its simplicity, the proposed metrics reduced the original datasets significantly without losing the information quality.

FUTURE WORK

- In the future work, the automatic setting of the threshold parameter, which has a crucial impact on the results, will be studied.

- A deeper insight into the parameter setting could be obtained by analysing the other, especially large-sized, UCI ML datasets.

Thank you for your time. Please join the discussion.



```
',Type=REAL_VALUED,Interval=[0,2.12199e-314],Attributes={}>
generation= 0
bestFitness= 0.854167 at 0 generation.
bestFitness= 0.895833 at 0 generation.
generation= 1
bestFitness= 0.916667 at 1 generation.
generation= 2
generation= 3
generation= 4
bestFitness= 0.9375 at 4 generation.
generation= 5
generation= 6
generation= 7
generation= 8
generation= 9
generation= 10
Total time= 0.546000
Writing file= 1, f_name= ../out/temp_uarmSolver.txt...
Rule(s) processed: 255

Damijan Novak@DESKTOP-DNOV ~/uARMSolver-master/bin
$
```