TinyNARM: Simplifying numerical association rule mining for running on microcontrollers

Iztok Fister Jr., Iztok Fister, Akemi Galvez, Andres Iglesias

September 16, 2023

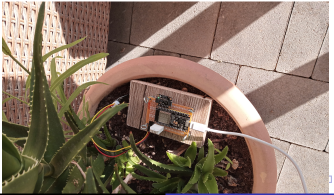
Agenda

Contributions of our study

- tinyNARM is an experimental effort in approaching/tailoring the classical Numerical Association Rule Mining to limited hardware devices.
- The tinyNARM reduces the number of attributes in the transaction database by discretizing the continuous numeric attributes.
- An evaluation of the proposed method.
- The benefit of using tinyNARM is outlined in smart agriculture.

Motivation

- Practical utilization of Numerical Association Rule Mining (NARM) in smart agriculture.
- We built a hardware system based on ESP32 for capturing data of plants, but the data mining part was done on an in-cloud solution.
- Why data mining on the same device?
  - Several rural areas do not have access to the internet.
  - Having direct access to data mining results, which can lead to potential actions, may be more efficient.



Classical NARM approaches

- Typically, the NARM was solved using Swarm Intelligence (SI) or Evolutionary Algorithms (EAs) due to the complexity of the problem.
- Both types belong to a family of stochastic nature-inspired population-based algorithms and differ from each other according to the principle of exploring the problem search space.
- NARM algorithms can deal with numerical and categorical attributes.

TinyML

- Modern AI-oriented applications significantly demand computing resources (high electricity consumption).
- Recently, we have witnessed the advent of compelling computer devices (e.g., mobile devices, IoT, single-board computers) that change the view of traditional computing
- The emergence of the new computational paradigm, i.e., tinyML. The tinyML proposes the so-called on-device learning that uses ML applications and model training on the device.

Proposed method

- The purpose of the tinyNARM is to reduce the continuous intervals of numeric attributes in the transaction database by introducing their discretization.
- The interval bears the characteristics of the whole numeric attribute calculated in the beginning.
- In the continuation, all the combinations C_r^M of n possible attributes by r selected ones are varied from $r = 2, \ldots, n$, where the implication sign is moved from the first to the last position in the combination of association rule in steps of one.
- Thus, the best association rule is searched for regarding the support.

Pseudocode of the tinyNARM

Require: $T = \langle o_1, \ldots, o_m \rangle$ ▷ Load transaction database
Ensure: $arch\_c$, $best\_c^*$ ▷ An archive and the best combination
for all $A^{(num)} \in D$ **do**
  Discretize $A_i^{(num)} = \{da_{i,j}\}$, where $d_{i,j} = \lfloor \frac{max\_o_i - min\_o_i}{N} \rfloor$
  Calculate frequencies $supp(da_{i,j}) = \frac{Freq(da_{i,j})}{N}$ for $i = 1, \ldots, M_i$
  Find the maximum element $A_i^{(dis)} = \max_{j=1,\ldots,M_i} supp(da_{i,j})$
  $D' = (D - A_i^{(num)}) \cup A^{(disc)}{}_i$
end for
for all $r \in [2, M-1]$ **do**
  for all $c \in C_r^M$ **do**
    $arch\_c \cup c_{cp}$ for $cp \in [1, r-1]$
    $best\_c = \max_{cp \in [1, r-1]} supp(c_{cp})$
  end for
  $best\_c^* = \max(best\_c^*, best\_c)$
end for

Experiments

- The main goal of the experiments was to evaluate how close to the results of the competitive NiaARM can approach the results of the proposed tinyNARM.
- The tinyNARM algorithm was coded in the Python programming language. Source code is available at: `https://gitlab.com/firefly-cpp/tinynarm`
- A part of the future work, we plan to run experiments on ESP32 microcontrollers, resulting in a new C language implementation of the tinyNARM based on the current Python prototype.

Datasets

Table: Datasets utilized in our study.

Dataset	Type	Features	Instances
Abalone	Categorical/Numerical	9	4,177
Breast_Cancer	Categorical	10	286
Nursery	Categorical	9	12,960
Wine	Numerical	14	178

Results

Algorithm	Measures	Abalone	Breast_Cancer	Nursery	Wine
tinyNARM	max	0.743	0.823	0.667	0.511
	min	0.558	0.554	0.222	0.503
	mean	0.623	0.649	0.274	0.505
	median	0.609	0.631	0.236	0.506
	std	0.059	0.081	0.101	0.003
	time	2.26	5.344	2.44	157.72
	no. rules	1,320	2,504	1,320	10,027
NiaARM	max	0.688	0.760	0.510	0.534
	min	0.549	0.579	0.503	0.511
	mean	0.605	0.638	0.505	0.519
	median	0.601	0.611	0.505	0.518
	std	0.041	0.057	0.002	0.007
	time	14.020	13.895	16.420	15.649
	no. rules	603	912	1,503	632

Conclusion

- The paper is a preliminary experimental study that presents how to move the in-cloud ML method to on-device tinyML, where the NARM was considered an ML method.
- Some drawbacks were observed, e.g., sometimes the exhaustive search can be even more time complex when many numeric attributes exist in the transaction database.
- Finding a new heuristic algorithm to cope with many numerical attributes in the database.
- New challenge: porting this Python implementation to C and running it on ESP32.