Discovering Association Rules in High-Dimensional Small Tabular Data

Erkan Karabulut[1,*], Daniel Daza[2], Paul Groth[1] and Victoria Degeler[1]

[1]University of Amsterdam, Science Park 900, 1098 XH Amsterdam, The Netherlands
[2]Amsterdam UMC location Vrije Universiteit Amsterdam, Department of Laboratory Medicine, De Boelelaan 1117, Amsterdam, the Netherlands

Abstract

Association Rule Mining (ARM) aims to discover patterns between features in datasets in the form of propositional rules, supporting both knowledge discovery and interpretable machine learning in high-stakes decision-making. However, in high-dimensional settings, rule explosion and computational overhead render popular algorithmic approaches impractical without effective search space reduction—challenges that propagate to downstream tasks. Neurosymbolic methods, such as Aerial+, have recently been proposed to address the rule explosion in ARM. While they tackle the high-dimensionality of the data, they also inherit limitations of neural networks, particularly reduced performance in low-data regimes.

This paper makes three key contributions to association rule discovery in high-dimensional tabular data. First, we empirically show that Aerial+ scales one to two orders of magnitude better than state-of-the-art algorithmic and neurosymbolic baselines across five real-world datasets. Second, we introduce the novel problem of ARM in high-dimensional, low data settings, such as gene expression data from the biomedicine domain with ~18K features and ~50 samples. Third, we propose two fine-tuning approaches to Aerial+ using tabular foundation models. Our proposed approaches are shown to significantly improve rule quality on five real-world datasets, demonstrating their effectiveness in low-data, high-dimensional scenarios.

Keywords

neurosymbolic ai, association rule mining, interpretable machine learning, tabular data

1. Introduction

Association Rule Mining (ARM) is the task of discovering patterns among the features of a dataset in the form of logical implications [1], also known as if-then rules. ARM has been applied in a myriad of domains for knowledge discovery [2] as well as for high-stakes decision-making as part of interpretable machine learning models [3, 4]. High-dimensional datasets, e.g., with thousands of columns, often lead to rule explosion and prolonged execution times [5]. Common solutions to rule explosion in ARM include constraining data features (i.e., ARM with item constraints [6, 7, 8]), mining top-k high-quality rules [9, 10], and closed itemset mining [11]. However, these methods mainly focus on reducing the search space for knowledge discovery, rather than directly addressing the computational burden of rule mining.

Neurosymbolic methods for ARM, such as Aerial+ [12], have been recently proposed to address the rule explosion problem on tabular data. Despite its effectiveness in addressing the rule explosion problem in generic tabular data, Aerial+ has not yet been evaluated on high d-dimensional datasets for scalability. Moreover, neurosymbolic methods for ARM also inherit the limitations of neural networks, such as reduced performance in low-data (n) regimes [13].

As is the case for several data-driven methods, Aerial+ relies on statistical patterns present in the dataset. In small datasets, such patterns may be hard to extract, which in turn may lead to reduced predictive performance, and in the case of Aerial+, to rules that do not accurately capture the true underlying patterns. Recent works on models for tabular data have addressed this issue by introducing *foundation models* [16, 17, 18, 19, 20], which are pre-trained on large datasets and transferred to small datasets without additional training, thereby providing strong inductive biases and generalizable representations that compensate for the limited data instances.

Table 1
Sample $d \gg n$ dataset and association rules. Gene expression datasets in tabular form often consist of 10K+ columns and a limited number of rows. This is a sample gene expression level data from [14], partially pre-processed by [15] and put in discrete form after applying z-score binning. Listed association rules are learned using Aerial+ [12] with item constraints on low and high gene expression levels.

Sample / Rule	Gene_1	Gene_2	Gene_3	⋯	Gene_18107
Sample_1	normal	normal	normal	⋯	normal
Sample_2	normal	normal	high	⋯	high
Sample_3	normal	normal	normal	⋯	low
		⋯			

Rule_1	Gene2 (high) ∧ Gene29 (high) → Gene14 (low)
Rule_2	Gene3 (high) ∧ Gene45 (high) → Gene84 (high)

In this paper, we make three key contributions to ARM research on categorical tabular data. First, we evaluate the scalability of both commonly used algorithmic ARM approaches as well as the recent Neurosymbolic methods on high d-dimensional datasets. Second, to the best of our knowledge, we introduce the problem of ARM on $d \gg n$ datasets for the first time, which are common in the biomedicine domain, such as gene expression datasets [14] (see Table 1), and evaluate the recent neurosymbolic methods on such datasets in terms of statistical rule quality and execution time. Third, we propose two fine-tuning methods for neurosymbolic ARM methods that rely on tabular foundation models for addressing the low-data regime.

Our empirical results show that: i) Aerial+ scales one to two orders of magnitude faster on high-dimensional datasets compared to state-of-the-art ARM methods (Section 3), ii) neurosymbolic methods need longer training to find high-quality association rules on $d \gg n$ datasets (Section 4), iii)

ANSyA 2025: 1st International Workshop on Advanced Neuro-Symbolic Applications, co-located with ECAI 2025.
*Corresponding author.
e.karabulut@uva.nl (E. Karabulut); d.f.dazacruz@amsterdamumc.nl (D. Daza); p.t.groth@uva.nl (P. Groth); v.o.degeler@uva.nl (V. Degeler)
0000-0003-2710-7951 (E. Karabulut); 0000-0002-5357-3705 (D. Daza); 0000-0003-0183-6910 (P. Groth); 0000-0001-7054-3770 (V. Degeler)

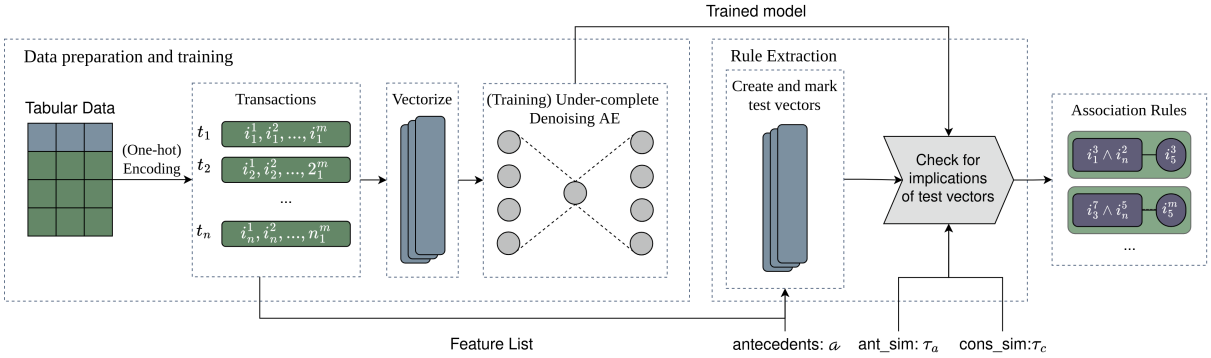

Figure 1: Aerial+ [12] ARM pipeline consists of: i) converting given categorical tabular data into transactions by one-hot encoding, ii) vectorizing the one-hot encoded data, iii) training an under-complete denoising Autoencoder with a reconstruction loss and to output probability distributions per column, iv) and extracts association rules by exploiting the reconstruction ability of autoencoders, based given probabilistic antecedent and consequent similarity thresholds.

our two proposed fine-tuning methods allow Aerial+ to learn significantly higher quality rules in small datasets (Section 4). The results indicate that neurosymbolic methods, especially when supported with tabular foundation models, can enable scalable and high-quality knowledge discovery in high-dimensional tabular data with few instances (Section 5).

2. Related Work

This section presents a formal definition of ARM on categorical tabular data, the problem of high-dimensional data with few instances, neurosymbolic ARM methods, and tabular foundation models.

Association rule mining. Following the original definition of ARM in [1], let $I = \{i_1, i_2, ..., i_m\}$ be a set of m items, and let $D = \{t_1, t_2, ..., t_n\}$ be a set of n transactions where $\forall t \in D, t \subseteq I$ meaning each transaction t consists of a set of items in I. An association rule is of the form $X \rightarrow Y$, where $X, Y \subseteq I$, is a first-order Horn clause with at most one positive literal, $|Y| = 1$ and $|X| \geq 1$, in its Conjunctive Normal Form (CNF) ($\neg X \vee Y$), and $X \cap Y = \emptyset$. Note that $p \rightarrow q \wedge r$ can be rewritten as $p \rightarrow q$ and $p \rightarrow r$, (i.e., $p, q, r \in I$). X is often referred to as the *antecedent* while Y is the *consequent* side of the association rule. Example association rules are given in Table 1. A rule $X \rightarrow Y$ is said to have *support* percentage s if s% of $t \in D$ contain $X \cup Y$, while the *confidence* of a rule is defined as $\frac{support(X \rightarrow Y)}{support(X)}$. ARM has initially been defined as the problem of finding rules that have higher minimum support and confidence values than a given user-defined threshold. The state-of-the-art in ARM literature has a plethora of sub-problems and solutions which can be found in [2, 21]. Categorical tabular data is often converted to a set of transactions via one-hot encoding, where each encoded value represents the presence (1) or absence (0) of a column-value pair, corresponding to items in I, and each row corresponds to a transaction in D.

ARM for high-dimensional small data. Having high-dimensional data with a limited number of samples is common in domains such as biomedicine, as in gene expression datasets [14] where there are 10K+ columns (different genes) and less than 100 rows (samples, e.g., patients). High-dimensionality of data has many solutions in the ARM literature, as it leads to rule explosion and, therefore, prolonged execution times. Existing methods include: i) mining rules for items of interest rather than all items, known as ARM with item constraints [6, 7, 8], ii) mining top-k high-quality rules based on a given rule quality criteria [9, 10] and, iii) reducing rule redundancy by identifying only frequent itemsets without frequent supersets of equal support, known as closed itemset mining [11]. Aerial+ [12] (and the earlier version Aerial [22]), is a neurosymbolic method that is orthogonal to many of the existing solutions and leverages neural networks to learn a concise set of high-quality rules with full data coverage. Despite showing promising results on generic tabular datasets, it has not yet been evaluated on high-dimensional data. Furthermore, we argue that using neural networks for ARM inherits neural network–specific issues, most notably reduced performance in low-data regimes [13]. To our knowledge, low-data scenarios in ARM have not yet been addressed, as employing neural networks for ARM represents a new paradigm shift.

Neurosymbolic methods for ARM. Neural networks have been used to mine association rules directly from tabular data in the past few years. Patel and Yadav [24] proposed the first approach that identifies frequent itemsets before constructing rules, but the work lacks an explicit algorithm or source code. Berteloot et al. [25] introduced ARM-AE, an autoencoder-based [26] method to mine association rules directly. Aerial+ [12] tackles the rule explosion problem in ARM by using an under-complete denoising autoencoder [27] to learn a compact data representation, and by introducing a more scalable extraction method than ARM-AE (Figure 1). This results in a smaller set of high-quality rules with full coverage over the data. Both Aerial+

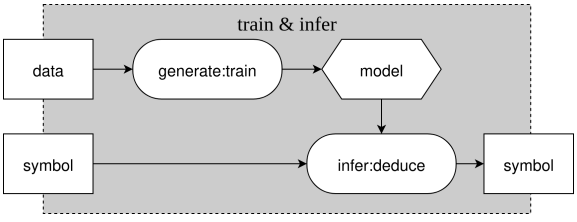

Figure 2: Boxology [23] diagram of neurosymbolic ARM approaches such as Aerial+: i) a neural *model* of *data* (i.e., tabular data) is learned, ii) an algorithm (symbolic) *infers* rules (symbols) from the model using hypotheses (symbols, as in test vectors of Aerial+ for probing the model).

and ARM-AE are neurosymbolic methods, combining neural models with symbolic rule extraction (Figure 2). However, Aerial+ has not yet been evaluated on high-dimensional datasets, which we address in this work.

Tabular foundation models are large neural networks pre-trained on vast collections of tabular data to capture table semantics and support diverse downstream tasks [28]. Among them, Tabular Prior-data Fitted Network (TabPFN) [16] is trained on millions of synthetic tables generated via structural causal models [29], and supports classification and regression. Other recent tabular foundation models include CARTE [30], which leverages graph-based representations trained on real-world knowledge graphs [31]; TabICL [17], which frames tabular learning as in-context learning; Tabbie [18], which uses masked token modeling for pretraining; and TableGPT [20], which adopts large language models for table understanding. Crucially, TabPFN is the only continuously maintained model that explicitly exposes an interface to extract table embeddings, which we utilized to develop fine-tuning strategies for Aerial+'s autoencoder architecture to learn higher-quality association rules in tables with a low number of rows [1].

3. ARM on high-dimensional tabular datasets

Given the focus on low-dimensional datasets in prior work on ARM, we begin with an empirical evaluation of the scalability of the state-of-the-art algorithmic and neurosymbolic ARM methods on high-dimensional categorical tabular datasets with few instances. Specifically, we aim to answer: *how does the runtime cost of current ARM methods scale in the case of high-dimensional datasets?*

Open-source. All the source code and datasets used in all the experiments can be found in https://github.com/DiTEC-project/rule_learning_high_dimensional_small_tabular_data.

Hardware. All experiments are run on a 12th Gen Intel® Core™ i5-1240P × 16 CPU, with 16 GiB memory, and 512 GB disk space. No GPUs were used, and no parallel execution was conducted.

Datasets. We use 5 $d \gg n$ gene expression datasets from [32, 33, 34, 14] (listed in Table 2), which are pre-processed according to the procedure described in [35] by [15]. The pre-processing consists of the trimmed mean of m-values normalization, log transformation (i.e., $log(x+1)$), and the expression values were made to have zero mean and unit standard deviation. Furthermore, to enable ARM on gene expression datasets, we applied z-score binning with one standard deviation as the cutoff to discretize values into high, low, and medium levels, as exemplified in Table 1.

Algorithms. We run the state-of-the-art neurosymbolic ARM method Aerial+ [12], commonly used algorithmic methods, ECLAT [36] and FP-Growth [37], as well as ARM-AE [25] on all the datasets given in Table 2. FP-Growth remains one of the most widely used ARM algorithms due to its efficiency and adaptability. Numerous variations to FP-Growth have been proposed to mitigate rule explosion and improve scalability, including Guided FP-Growth [38] for item-constrained mining, parallel FP-Growth [39], and GPU-accelerated versions [40] for faster execution. Note that Aerial+ also supports item constraints, parallel, and GPU executions. However, we only compare the basic version of each algorithm.

[1]We rely on the implementation available at https://github.com/PriorLabs/TabPFN.

Table 2
High-dimensional tabular gene expression datasets with few instances, used in all experiments [32, 33, 34, 14].

Dataset	# Columns	# Rows
Chondrosarcoma	18006	6
SmallCellLungCarcinoma	18237	60
NonSmallCellLungCarcinoma	18108	86
BreastCarcinoma	18061	51
Melanoma	17902	55

Table 3
Evaluated algorithms and hyperparameters for fair ARM comparison on high-dimensional, low-sample tabular data (R = Aerial+ rules, C = Columns).

Algorithm	Type	Parameters
Aerial+	Neurosymbolic	$a = 2, \tau_a = 0.5, \tau_c = 0.8$
ARM-AE	Neurosymbolic	M=2, N=$\lvert R\rvert/\lvert C\rvert$, L=0.5
FP-Growth	Algorithmic	antecedents = 2, min_conf=0.8,
ECLAT	Algorithmic	min_support=0.5 $*$ $\mathbb{E}$[support(R)]

Experimental setup and hyperparameters. To ensure a fair comparison, we set the hyperparameters of each method (shown in Table 3) as follows: i) number of antecedents is set to 2 for all methods, ii) Aerial+'s antecedent similarity threshold (τ_a) and ARM-AE's likeness (L) are set to 0.5, iii) Aerial+'s consequent similarity threshold (τ_c) and minimum confidence of the algorithmic methods are set to 0.8, iv) minimum support threshold of the algorithmic methods are set to half the average support of the rules learned by Aerial+, to ensure comparable average support values, v) ARM-AE's number of rules per consequent (N) is set to Aerial+'s rule count divided by the number of columns to ensure comparable rule counts, vi) and both Aerial+ and ARM-AE were trained for 10 epochs with a batch size of 2. Aerial+ is implemented using the pyAerial [2] [41] library, FP-Growth is implemented using MLxtend [42], ECLAT is implemented using pyECLAT [3], and ARM-AE is implemented using its original repository [4]. The **goal** of this experimental setup is to test the scalability of the algorithms, and not to perform a rule quality comparison, which has already been done in earlier work [12].

Results. Figure 3 shows the execution time of each method, in seconds on a logarithmic scale, on 5 datasets as the number of columns increases. Execution times include both training and the rule extraction times for the neurosymbolic methods. The results show that Aerial+ has one to two orders of magnitude faster execution times than the other methods. The gap in execution time increases as the number of columns increases. We also see that the algorithmic method FP-Growth runs faster when the number of columns is smaller than 30. This shows that Aerial+'s training is only compensated if the tables have more than 30 columns. Note that Aerial+ has linear time complexity during training and polynomial time over the number of columns (after one-hot encoding) during the extraction.

[2]https://github.com/DiTEC-project/pyaerial
[3]https://github.com/jeffrichardchemistry/pyECLAT
[4]https://github.com/TheophileBERTELOOT/ARM-AE/tree/master

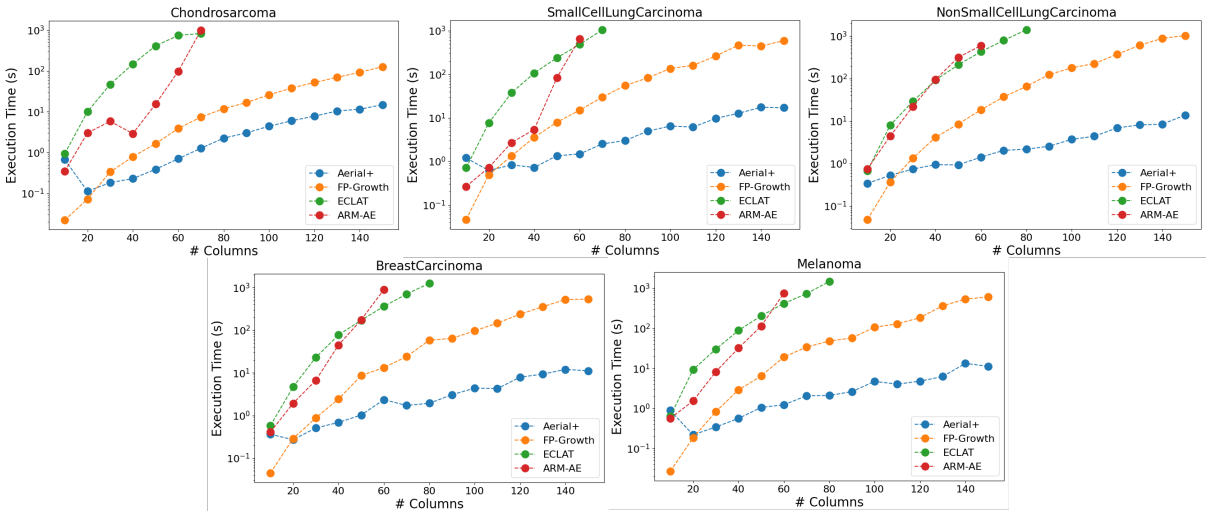

Figure 3: Scalability on high-dimensional tabular data. Execution times of algorithmic and neurosymbolic (including training and rule extraction time) ARM approaches in seconds on a logarithmic scale, as the number of columns increases gradually. Aerial+ has one to two orders of magnitude better scalability on high-dimensional datasets compared to other methods. Lower performance of Aerial+ with a smaller number of columns is due to the training procedure, which implies that algorithmic methods are faster on lower-dimensional (columns) tables.

4. Neurosymbolic ARM in low-data regime

Experiments in Section 3 showed that the fastest algorithmic solution, FP-Growth, takes ~10^3 seconds on tables with only 150 columns and 2 antecedents, while a neurosymbolic method, Aerial+, runs one to two orders of magnitude faster. This empirically validates the scalability of neurosymbolic approaches to ARM. However, we argue that Aerial+ also inherits the known issues in neural networks, particularly the decline in performance in a low-data regime [13]. Concretely, Aerial+ relies on training a deep autoencoder on the tabular data with a reconstruction objective. Following results from statistical learning theory [43] and empirical observations in neural networks [44], this implies that Aerial+'s performance is bounded by the number of training samples, and with small data it may yield rules that do not accurately capture ground-truth associations.

An effective approach for addressing data scarcity is *transfer learning* [45], which requires training a neural network, or vector representations (i.e. *embeddings*) on a large dataset, that then can be transferred to a downstream task on a small dataset. This provides a starting point that can improve performance in comparison to learning from scratch on a small dataset.

In this work, we propose two fine-tuning strategies to Aerial+ using TabPFN [16], a foundation model for tabular data that has been pre-trained over millions of tables, which we use to generate embeddings for the small datasets in our experiments.

4.1. Fine Tuning with Pre-trained Weight Initialization

Figure 4 illustrates the fine-tuning strategy introduced in this section (Aerial+WI). On a high level, table embeddings from a tabular foundation model are utilized to initialize the weights of Aerial+'s under-complete denoising autoencoder, providing a semantically meaningful starting point for learning compact data representations.

Let $X \in \mathbb{R}^{n \times d}$ denote the tabular dataset and $y \in \mathbb{R}^n$ the corresponding labels. We first compute fixed-length embeddings for each row in X using a pretrained TabPFNClassifier. These embeddings, denoted as $E \in \mathbb{R}^{n \times d_e}$, where d_e is the embedding dimension, are generated via a 10-fold TabPFN-based meta-learning scheme:

$$E = f_{\text{TabPFNClassifier}}(X, y)$$

We then one-hot encode X into $\hat{X} \in \mathbb{R}^{n \times d'}$ following the original Aerial+ pipeline, where d' is the total number of binary features after encoding categorical attributes. A two-layer projection encoder $g_\theta : \mathbb{R}^{d'} \to \mathbb{R}^{d_e}$ is trained to map $\hat{X}$ to the TabPFN embedding space. The encoder architecture is as follows:

$$g_\theta(\hat{X}) = W_2 \cdot \text{Dropout}(\sigma(\text{LayerNorm}(W_1 \hat{X} + b_1))) + b_2$$

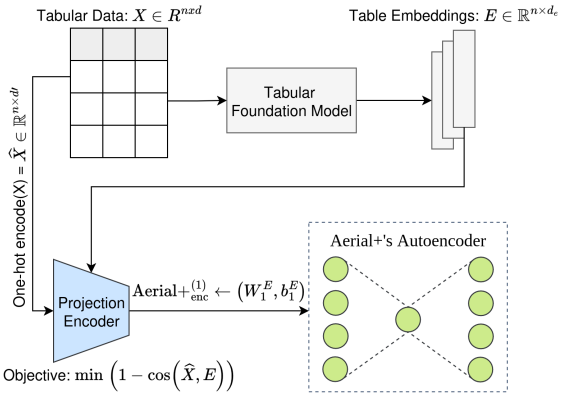

Figure 4: Fine tuning with pre-trained weight initialization (Aerial+WI): i) tabular data is embedded via a tabular foundation model, ii) a projection encoder is trained to align table embeddings with pre-processed Aerial+ input tabular data using a cosine loss objective, iii) and first-layer weights and biases of Aerial+'s encoder is initialized via the projection encoder, providing a semantically meaningful starting point.

where $W_1 \in \mathbb{R}^{h \times d'}$, $W_2 \in \mathbb{R}^{d_e \times h}$, h is the hidden dimension, σ is the LeakyReLU activation with a negative slope of 0.01, and LayerNorm and Dropout ($p = 0.1$) are applied for regularization.

The projection encoder is trained using a cosine loss function to align $g_\theta(\hat{X})$ with E:

$$\mathscr{L}(\theta) = 1 - \frac{1}{n} \sum_{i=1}^{n} \cos\left(g_\theta(\hat{x}_i), E_i\right)$$

Training is performed using Adam optimizer [46] for 25 epochs with early stopping if the validation loss plateaus (with early stopping patience of 20 and a minimum improvement threshold of 10^{-4}). After training, the weight matrix W_1 and bias b_1 from the first layer of g_θ are used to initialize the corresponding parameters in the first layer of Aerial+'s encoder:

$$\text{Aerial+}^{(1)}_{\text{enc}} \leftarrow (W_1, b_1)$$

This initialization provides a strong inductive prior for Aerial+, guiding its encoder to start from a semantically meaningful representation space derived from TabPFN's meta-learned embeddings.

Note that the gene expression datasets contain no predefined class labels. Therefore, a random column is selected as the target variable to enable TabPFN embedding generation.

4.2. Projection-Guided Fine Tuning via Double Loss

Figure 5 visualizes the fine-tuning strategy described in this section (Aerial+DL). Conceptually, this strategy uses a projection encoder to align Aerial+ reconstructions with table embeddings from a tabular foundation model, jointly optimizing reconstruction and alignment losses for semantic consistency.

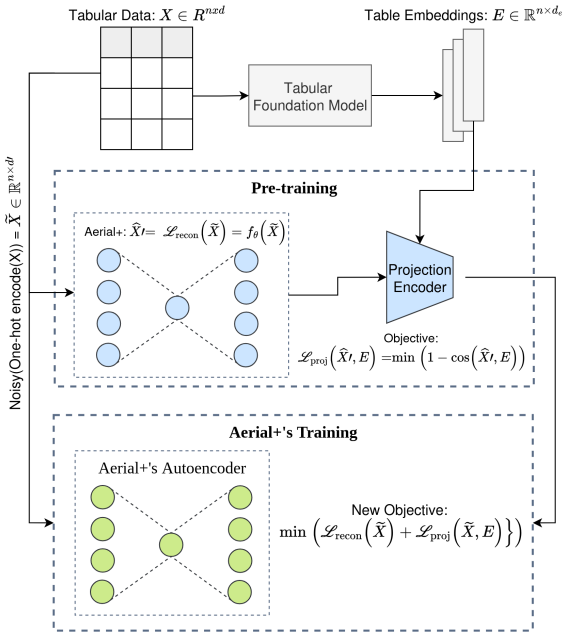

Figure 5: Projection-Guided Fine-Tuning via Double Loss (Aerial+DL): i) tabular data is embedded via a tabular foundation model, ii) a projection encoder is trained to align table embeddings with reconstructed Aerial+ output of the tabular data, using a cosine loss, iii) during Aerial+ autoencoder training, a new objective of aligning projection encoder output with the table embeddings is added to the reconstruction loss, supporting semantic alignment of the autoencoder reconstruction process to the table embeddings.

Building on the projection encoder g_θ described in Section 4.1, this second fine-tuning strategy aligns the Aerial+'s autoencoder reconstructions with TabPFN embeddings using a double loss function.

Unlike the first strategy, where g_θ was trained directly on raw one-hot inputs, here we first pass a corrupted version of the one-hot input $\hat{X}$ through Aerial+'s initial autoencoder f_θ and train g_θ on its outputs. Specifically, we generate noisy inputs (following the same strategy as Aerial+):

$$\tilde{X} = \text{clip}(\hat{X} + \epsilon), \quad \epsilon \sim \mathcal{N}(0, \sigma^2)$$

where $\sigma = 0.5$ and values are clipped to $[0, 1]$. We then compute reconstructions $\hat{X}' = f_\theta(\tilde{X})$. The projection encoder is trained to map these reconstructions to their corresponding TabPFN embeddings $E \in \mathbb{R}^{n \times d_e}$:

$$g_\theta(\hat{x}_i') \approx E_i$$

by minimizing the cosine distance loss:

$$\mathscr{L}_{\text{proj}}(\theta) = 1 - \frac{1}{n} \sum_{i=1}^{n} \cos(g_\theta(\hat{x}_i'), E_i)$$

After this pretraining phase, g_θ is frozen, and Aerial+'s autoencoder is fine-tuned using a *double loss* objective:

$$\mathscr{L}(\theta) = \mathscr{L}_{\text{recon}}(f_\theta(\tilde{x}), \hat{x}) + \mathscr{L}_{\text{proj}}(g_\theta(f_\theta(\tilde{x})), E)$$

where $\mathscr{L}_{\text{recon}}$ is a binary cross-entropy loss applied to per one-hot encoded column value as in Aerial+, generating probability distributions per column. The double loss strategy encourages Aerial+'s autoencoder to not only reconstruct the original data, but also to produce representations that are semantically consistent with TabPFN's meta-learned embedding space.

4.3. Experimental Results

Setup and hyperparameters. We run Aerial+ and the two fine-tuned versions, with pre-trained weight initialization (Aerial+WI) and double loss (Aerial+DL), on 5 $d \gg n$ datasets with 100 columns and compare their rule quality. The default Aerial+ uses Xavier [47] weight initialization as in the original work. All the approaches are run with 2 antecedents, for 25 epochs with a batch size of 2. Aerial+'s autoencoder for both the default and the fine-tuned versions consists of 2 layers per encoder and decoder, with the dimensions $\hat{X} \rightarrow 50 \rightarrow 10$, and the mirrored version for the decoder. We run each method **50 times** and present the average rule quality results for robustness.

Evaluation criteria. The standard rule quality metrics from the ARM literature are used as the evaluation criteria [2, 21]. Let D be a set of transactions as introduced in Section 2, and $R = \{r_1, r_2, ..., r_t\}$ be the rule set learned by each approach where $\forall r_i \in R, r_i = (X_i \rightarrow Y_i)$:

- **Number of rules.** Total number of rules learned:$|R|$
- **Average rule coverage.** Average number of transactions where the rule antecedent appears: $\text{AvgCov} = \frac{1}{|R|} \sum_{i=1}^{|R|} |\{t \in D \mid X_i \subseteq t\}|$

Table 4
Rule quality of Aerial+ in low-data regime. Fine-tuning Aerial+ with the weight initialization (Aerial+WI) and double loss (Aerial+DL) methods based on TabPFN embeddings consistently outperformed the default version in rule confidence and association strength (Zhang's). Fine-tuning produced fewer rules with lower data coverage on 3 of 5 datasets, as expected due to the elimination of relatively obvious (low-association-strength) rules. Execution time increased by only a few seconds, which is negligible in a low-data regime.

Approach	# Rules	~Rule Coverage	~Support	~Confidence	Data Coverage	~Zhang's Metric	Exec. Time (s)
Chondrosarcoma							
Aerial+	200	**0.23**	0.21	0.921	0.533	0.784	2.25
Aerial+WI	75	0.217	0.206	0.945	0.524	0.813	5.80
Aerial+DL	75	0.235	0.219	**0.947**	**0.536**	**0.828**	5.36
SmallCellLungCarcinoma							
Aerial+	1576	0.068	0.041	0.579	**0.835**	0.476	10.58
Aerial+WI	664	**0.076**	0.052	**0.633**	0.715	**0.577**	13.48
Aerial+DL	1338	0.070	0.044	0.597	0.816	0.513	18.23
NonSmallCellLungCarcinoma							
Aerial+	1620	0.059	0.035	0.584	0.823	0.554	18.03
Aerial+WI	978	**0.078**	0.057	**0.663**	0.698	**0.639**	28.67
Aerial+DL	1453	0.053	0.028	0.547	**0.849**	0.501	24.27
BreastCarcinoma							
Aerial+	1017	0.072	0.046	0.641	**0.816**	0.575	9.64
Aerial+WI	590	0.077	0.052	**0.686**	0.686	**0.644**	12.09
Aerial+DL	535	**0.078**	0.050	0.652	0.761	0.590	15.31
Melanoma							
Aerial+	1220	0.067	0.035	0.545	**0.888**	0.440	13.09
Aerial+WI	773	0.070	0.038	**0.575**	0.772	**0.496**	13.19
Aerial+DL	859	**0.071**	0.038	0.566	0.860	0.461	16.49

- **Average support.** Average fraction of transactions containing both antecedent and consequent:

$$\text{AvgSupp} = \frac{1}{|R|}\sum_{i=1}^{|R|}\frac{|\{t \in D \mid X_i \cup Y_i \subseteq t\}|}{|D|}$$

- **Average confidence.** Average conditional probability that the consequent appears given the antecedent:

$$\text{AvgConf} = \frac{1}{|R|}\sum_{i=1}^{|R|}\frac{|\{t \in D \mid X_i \cup Y_i \subseteq t\}|}{|\{t \in D \mid X_i \subseteq t\}|}$$

- **Total data coverage.** Fraction of transactions covered by at least one rule antecedent:

$$\text{TotalCov} = \frac{\left|\bigcup_{i=1}^{|R|}\{t \in D \mid X_i \subseteq t\}\right|}{|D|}$$

- **Average Zhang's metric [48].** Average statistical dependence between antecedent and consequent beyond chance:

$$\text{AvgZhang} = \frac{1}{|R|}\sum_{i=1}^{|R|}\text{Zhang}(X_i \Rightarrow Y_i)$$

where:

$$\text{Zhang}(X_i \to Y_i) = \frac{\text{conf}(X_i \to Y_i) - \text{conf}(X_i^{'} \to Y_i)}{max(\text{conf}(X_i \to Y_i), \text{conf}(X_i^{'} \to Y_i))}$$

with conf being the confidence score of a rule and $X_i^{'}$ referring to the absence of X_i in D.

- **Execution time.** Sum of model training time, fine-tuning (when applicable), and rule extraction time in seconds.

Results. Table 4 shows the rule quality evaluation results of Aerial+ and the two fined-tuned versions Aerial+WI and Aerial+DL on 5 datasets. The results show that Aerial+WI outperforms Aerial+ in terms of rule confidence and association strength (Zhang's metric) on all 5 datasets. Aerial+DL's confidence and association strength also exceed Aerial+'s on 4 out of 5 datasets, except the NonSmallCellLungCarcinoma dataset. Both fine-tuning methods resulted in a smaller number of rules on all datasets and with a smaller data coverage on 3 out of the 5. This is expected as the fine-tuned versions capture rules with higher association strength on average, meaning the less obvious rules are eliminated during the rule extraction process, and therefore, the final data coverage was lower. The fine-tuned methods have higher support values on 4 out of 5 datasets. However, we do not take the high support values as a positive sign, as it depends on the application. For instance, high support rules are good at explaining trends in the data, while low support rules can be better at explaining anomalies. Lastly, fine-tuning resulted in only a few seconds of increment in the execution time, which is negligible in the low-data regime. Note that the costliest operation in Aerial+ is the rule extraction process and not the training (or pre-training), which is not significantly affected by the fine-tuning methods.

5. Discussion

The section discusses the experimental results, the role of neurosymbolic methods, and tabular foundation models in ARM.

Neurosymbolic methods scale better on high-dimensional data. Experiments in Section 3 show that Aerial+, a neurosymbolic method to ARM, has execution speed of one to two orders of magnitude faster than the

algorithmic ARM approaches. We argue this is because Aerial+ leverages neural networks' ability to handle high-dimensional data, it has linear complexity over the number of rows in training, and polynomial time complexity over the number (one-hot encoded) columns during the rule extraction stage. Algorithmic methods, on the other hand, rely on counting the co-occurrences of itemsets in the data, which is a costlier operation.

Aerial+ inherits neural networks-specific issues into ARM. The scalability of Aerial+ on high-dimensional data comes at a cost, most notably the reduced performance in the low-data regime for ARM. The original paper of Aerial+ trains only for 2 epochs on generic tabular datasets and was able to obtain high-quality rules. In the low-data regime, however, we were able to get high-quality rules consistently in each execution only after training for 25 epochs. This shows that while the neurosymbolic methods can help in scalability, they also introduce a new research problem into the ARM literature, namely, rule mining in the low-data regime.

Fine-tuning Aerial+ for better knowledge discovery. Experiments in Section 4 showed that our two proposed fine-tuning methods using the tabular foundation model TabPFN resulted in significantly higher-quality rules in comparison to the default version of Aerial+ on 5 real-world high-dimensional tabular datasets with few instances. Many of the other tabular foundation models that we investigated, including Tabbie, CARTE, TableGPT, and TabICL, do not provide an interface to obtain table embeddings. Therefore, we were not able to use them in our experiments. Since TabPFN is trained to perform classification and regression tasks over tabular data, we expect that models explicitly trained to learn column embeddings and associations could potentially result in better rule quality.

Neurosymbolic methods start a paradigm shift in ARM. We show that the Neurosymbolic ARM methods can be supported by prior-data fitted networks, as in TabPFN, to learn higher-quality rules. This raises the research question of **what other types of prior data or background knowledge can be utilized as part of ARM?** We invite researchers to further investigate neurosymbolic methods for ARM, as the neurosymbolic integration brings an immense potential for both knowledge discovery and fully interpretable inference across a plethora of domains.

Further validation of our approach and limitations. The algorithmic methods strictly depend on the distribution of data when mining rules in terms of execution time, as *denser* datasets where many frequent itemsets of high support are present will eventually prolong the execution time. Aerial+, however, applies the exact same polynomial-time rule extraction process regardless of the density of the data, and therefore depends less on the dataset attributes. However, we will still test our fine-tuning approaches on more datasets from diverse domains to further validate our approach in future work. Furthermore, we will evaluate our approach on generic tabular data with higher numbers of instances, i.e., $n \gg d$, to see whether it leads to early convergence or higher quality rules. Our proposed fine-tuning strategies are currently limited to the only available tabular foundation model with an explicit table embedding interface, TabPFN. Since TabPFN is specifically trained for classification and regression, this limitation may restrict performance improvements, and a future foundation model trained to capture column associations explicitly could significantly improve rule discovery.

6. Conclusions

This paper highlights the potential of neurosymbolic methods in the domain of association rule mining (ARM), especially under high-dimensional and low-sample ($d \gg n$) settings common in domains such as biomedicine. We have empirically shown that Aerial+, a neurosymbolic approach, offers substantial scalability improvements compared to the state-of-the-art neurosymbolic and algorithmic ARM techniques, scaling one to two orders of magnitude faster. However, neurosymbolic ARM also inherits the known issues of neural networks into ARM literature, specifically the reduced performance in low-data regimes, which we addressed through two targeted fine-tuning strategies.

Our fine-tuning methods use table embeddings from TabPFN, a tabular foundation model, to i) initialize the weights of Aerial+ (Aerial+WI), ii) and to better semantically align Aerial+ autoencoder training with a given tabular data (Aerial+DL). The results show that both Aerial+WI and Aerial+DL methods significantly improved rule quality in low-data settings. This demonstrates the promising role of pretrained tabular models in enhancing knowledge discovery over tabular datasets besides classification and regression tasks that are commonly tackled in the tabular data domain.

Looking forward, we see this as the beginning of a broader paradigm shift in ARM, where background knowledge and pretrained models can be explicitly leveraged to guide rule extraction. We invite the community to explore what other forms of prior knowledge, architectures, or foundation models can be integrated into neurosymbolic ARM. Future work will also validate our methods across a wider range of datasets and evaluate their effectiveness in high-instance scenarios ($n \gg d$), with the aim of achieving both scalability and high interpretability in real-world data mining applications.

Acknowledgements

This work has received support from the Dutch Research Council (NWO), in the scope of the Digital Twin for Evolutionary Changes in water networks (DiTEC) project, file number 19454.

Declaration on Generative AI

During the preparation of this work, the authors used ChatGPT (GPT-4.1) for paraphrasing and rewording. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] R. Agrawal, R. Srikant, et al., Fast algorithms for mining association rules, in: Proceedings of the 20th International Conference on Very Large Data Bases, VLDB, volume 1215, 1994, pp. 487–499.

[2] J. M. Luna, P. Fournier-Viger, S. Ventura, Frequent itemset mining: A 25 years review, WIREs Data Mining and Knowledge Discovery 9 (2019) e1329. doi:10.1002/widm.1329.

[3] C. Rudin, Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead, Nature machine intelligence 1 (2019) 206–215.

[4] E. Angelino, N. Larus-Stone, D. Alabi, M. Seltzer, C. Rudin, Learning certifiably optimal rule lists for categorical data, Journal of Machine Learning Research 18 (2018) 1–78.

[5] S. Moens, E. Aksehirli, B. Goethals, Frequent itemset mining for big data, in: 2013 IEEE international conference on big data, IEEE, 2013, pp. 111–118.

[6] R. Srikant, Q. Vu, R. Agrawal, Mining association rules with item constraints., in: Kdd, volume 97, 1997, pp. 67–73.

[7] E. Baralis, L. Cagliero, T. Cerquitelli, P. Garza, Generalized association rule mining with constraints, Information Sciences 194 (2012) 68–84.

[8] Z. Yin, W. Gan, G. Huang, Y. Wu, P. Fournier-Viger, Constraint-based sequential rule mining, in: 2022 IEEE 9th International Conference on Data Science and Advanced Analytics (DSAA), IEEE, 2022, pp. 1–10.

[9] P. Fournier-Viger, C.-W. Wu, V. S. Tseng, Mining top-k association rules, in: Advances in Artificial Intelligence: 25th Canadian Conference on Artificial Intelligence, Canadian AI 2012, Toronto, ON, Canada, May 28-30, 2012. Proceedings 25, Springer, 2012, pp. 61–73.

[10] L. T. Nguyen, B. Vo, L. T. Nguyen, P. Fournier-Viger, A. Selamat, Etarm: an efficient top-k association rule mining algorithm, Applied Intelligence 48 (2018) 1148–1160.

[11] M. J. Zaki, C.-J. Hsiao, Charm: An efficient algorithm for closed itemset mining, in: Proceedings of the 2002 SIAM international conference on data mining, SIAM, 2002, pp. 457–473.

[12] E. Karabulut, P. Groth, V. Degeler, Neurosymbolic association rule mining from tabular data, in: Proceedings of The 19th International Conference on Neurosymbolic Learning and Reasoning, volume 284 of *Proceedings of Machine Learning Research*, PMLR, 2025, pp. 565–588. URL: https://proceedings.mlr.press/v284/karabulut25a.html.

[13] B. Liu, Y. Wei, Y. Zhang, Q. Yang, Deep neural networks for high dimension, low sample size data., in: IJCAI, volume 2017, 2017, pp. 2287–2293.

[14] H. Gao, J. M. Korn, S. Ferretti, J. E. Monahan, Y. Wang, M. Singh, C. Zhang, C. Schnell, G. Yang, Y. Zhang, et al., High-throughput screening using patient-derived tumor xenografts to predict clinical trial drug response, Nature medicine 21 (2015) 1318–1325.

[15] C. Ruiz, H. Ren, K. Huang, J. Leskovec, High dimensional, tabular deep learning with an auxiliary knowledge graph, Advances in Neural Information Processing Systems 36 (2023) 26348–26371.

[16] N. Hollmann, S. Müller, L. Purucker, A. Krishnakumar, M. Körfer, S. B. Hoo, R. T. Schirrmeister, F. Hutter, Accurate predictions on small data with a tabular foundation model, Nature 637 (2025) 319–326.

[17] J. Qu, D. HolzmÃžller, G. Varoquaux, M. L. Morvan, Tabicl: A tabular foundation model for in-context learning on large data, arXiv preprint arXiv:2502.05564 (2025).

[18] H. Iida, D. Thai, V. Manjunatha, M. Iyyer, Tabbie: Pretrained representations of tabular data, arXiv preprint arXiv:2105.02584 (2021).

[19] P. Yin, G. Neubig, W.-t. Yih, S. Riedel, Tabert: Pretraining for joint understanding of textual and tabular data, arXiv preprint arXiv:2005.08314 (2020).

[20] A. Su, A. Wang, C. Ye, C. Zhou, G. Zhang, G. Chen, G. Zhu, H. Wang, H. Xu, H. Chen, et al., Tablegpt2: A large multimodal model with tabular data integration, arXiv preprint arXiv:2411.02059 (2024).

[21] M. Kaushik, R. Sharma, I. Fister Jr, D. Draheim, Numerical association rule mining: a systematic literature review, arXiv preprint arXiv:2307.00662 (2023).

[22] E. Karabulut, P. Groth, V. Degeler, Learning semantic association rules from internet of things data, Neurosymbolic Artificial Intelligence 1 (2025). doi:10.1177/29498732251377518.

[23] M. van Bekkum, M. de Boer, F. van Harmelen, A. Meyer-Vitali, A. t. Teije, Modular design patterns for hybrid learning and reasoning systems: a taxonomy, patterns and use cases, Applied Intelligence 51 (2021) 6528–6546.

[24] H. K. Patel, K. Yadav, An innovative approach for association rule mining in grocery dataset based on non-negative matrix factorization and autoencoder, Journal of Algebraic Statistics 13 (2022) 2898–2905.

[25] T. Berteloot, R. Khoury, A. Durand, Association rules mining with auto-encoders, in: International Conference on Intelligent Data Engineering and Automated Learning, Springer, 2024, pp. 51–62.

[26] D. Bank, N. Koenigstein, R. Giryes, Autoencoders, Machine learning for data science handbook: data mining and knowledge discovery handbook (2023) 353–374.

[27] P. Vincent, H. Larochelle, Y. Bengio, P.-A. Manzagol, Extracting and composing robust features with denoising autoencoders, in: Proceedings of the 25th international conference on Machine learning, 2008, pp. 1096–1103.

[28] Y. Ruan, X. Lan, J. Ma, Y. Dong, K. He, M. Feng, Language modeling on tabular data: A survey of foundations, techniques and evolution, arXiv preprint arXiv:2408.10548 (2024).

[29] J. Pearl, Causality, Cambridge university press, 2009.

[30] M. J. Kim, L. Grinsztajn, G. Varoquaux, Carte: pretraining and transfer for tabular learning, arXiv preprint arXiv:2402.16785 (2024).

[31] A. Hogan, E. Blomqvist, M. Cochez, C. d'Amato, G. D. Melo, C. Gutierrez, S. Kirrane, J. E. L. Gayo, R. Navigli, S. Neumaier, et al., Knowledge graphs, ACM Computing Surveys (Csur) 54 (2021) 1–37.

[32] W. Yang, J. Soares, P. Greninger, E. J. Edelman, H. Lightfoot, S. Forbes, N. Bindal, D. Beare, J. A. Smith, I. R. Thompson, et al., Genomics of drug sensitivity in cancer (gdsc): a resource for therapeutic biomarker discovery in cancer cells, Nucleic acids research 41 (2012) D955–D961.

[33] F. Iorio, T. A. Knijnenburg, D. J. Vis, G. R. Bignell, M. P. Menden, M. Schubert, N. Aben, E. Gonçalves, S. Barthorpe, H. Lightfoot, et al., A landscape of pharmacogenomic interactions in cancer, Cell 166 (2016) 740–754.

[34] M. J. Garnett, E. J. Edelman, S. J. Heidorn, C. D. Greenman, A. Dastur, K. W. Lau, P. Greninger, I. R. Thompson, X. Luo, J. Soares, et al., Systematic identification of genomic markers of drug sensitivity in cancer cells, Nature 483 (2012) 570–575.

[35] S. M. Mourragui, M. Loog, D. J. Vis, K. Moore, A. G. Manjon, M. A. van de Wiel, M. J. Reinders, L. F. Wessels, Predicting patient response with models trained

on cell lines and patient-derived xenografts by nonlinear transfer learning, Proceedings of the National Academy of Sciences 118 (2021) e2106682118.
[36] M. J. Zaki, S. Parthasarathy, M. Ogihara, W. Li, et al., New algorithms for fast discovery of association rules., in: KDD, volume 97, 1997, pp. 283–286.
[37] J. Han, J. Pei, Y. Yin, Mining frequent patterns without candidate generation, ACM sigmod record 29 (2000) 1–12.
[38] L. Shabtay, P. Fournier-Viger, R. Yaari, I. Dattner, A guided fp-growth algorithm for mining multitude-targeted item-sets and class association rules in imbalanced data, Information Sciences 553 (2021) 353–375.
[39] H. Li, Y. Wang, D. Zhang, M. Zhang, E. Y. Chang, Pfp: parallel fp-growth for query recommendation, in: Proceedings of the 2008 ACM conference on Recommender systems, 2008, pp. 107–114.
[40] H. Jiang, H. Meng, A parallel fp-growth algorithm based on gpu, in: 2017 IEEE 14th International Conference on e-Business Engineering (ICEBE), IEEE, 2017, pp. 97–102.
[41] E. Karabulut, P. Groth, V. Degeler, Pyaerial: Scalable association rule mining from tabular data, SoftwareX 31 (2025) 102341. doi:10.1016/j.softx.2025.102341.
[42] S. Raschka, Mlxtend: Providing machine learning and data science utilities and extensions to python's scientific computing stack, The Journal of Open Source Software 3 (2018). doi:10.21105/joss.00638.
[43] V. Vapnik, Statistical learning theory, Wiley, 1998.
[44] C. Zhang, S. Bengio, M. Hardt, B. Recht, O. Vinyals, Understanding deep learning (still) requires rethinking generalization, Commun. ACM 64 (2021) 107–115. doi:10.1145/3446776.
[45] K. R. Weiss, T. M. Khoshgoftaar, D. Wang, A survey of transfer learning, J. Big Data 3 (2016) 9. doi:10.1186/S40537-016-0043-6.
[46] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980 (2014).
[47] X. Glorot, Y. Bengio, Understanding the difficulty of training deep feedforward neural networks, in: Proceedings of the thirteenth international conference on artificial intelligence and statistics, JMLR Workshop and Conference Proceedings, 2010, pp. 249–256.
[48] X. Yan, C. Zhang, S. Zhang, Confidence metrics for association rule mining, Applied Artificial Intelligence 23 (2009) 713–737.