DiffVersify: a scalable approach to differentiable pattern mining with coverage regularization

Thibaut Chataing[1,2], Julien Perez[3], Marc Plantevit[3], and Céline Robardet[2]

[1] PALO IT, Lyon, France
[2] INSA Lyon, CNRS, LIRIS UMR 5205, F-69621 Villeurbanne, France
[3] EPITA Research Laboratory (LRE), FR-94276, Le Kremlin-Bicêtre, France

Abstract. Pattern mining addresses the challenge of automatically identifying interpretable and discriminative patterns within data. Recent approaches, leveraging differentiable approach through neural autoencoder with class recovery, have achieved encouraging results but tend to fall short as the magnitude of the noise and the number of underlying features increase in the data. Empirically, one can observe that the number of discovered patterns tend to be limited in these challenging contexts. In this article, we present a differentiable binary model that integrates a new regularization technique to enhance pattern coverage. Besides, we introduce an innovative pattern decoding strategy taking advantage of non-negative matrix factorization (NMF), extending beyond conventional thresholding methods prevalent in existing approaches. Experiments on four real-world datasets exhibit superior performances of DiffVersify in terms of the ROC-AUC metric. On synthetic data, we observe an increase in the similarity between the discovered patterns and the ground truth. Finally, using several metrics to finely evaluate the quality of the patterns in regard to the data, we show the global effectiveness of the approach.

1 Introduction

Pattern mining is a crucial field for extracting meaningful and easily interpretable insights from data. Traditional frequent pattern mining techniques [24], while widely used, often fail to capture all the underlying regularities in the data and tend to produce results that are overly general and redundant. To address this limitation, various techniques [6,10] have emerged, aimed at identifying a smaller yet more informative set of patterns. However, these approaches are computationally intensive due to the use of enumeration-based strategies on large search space. Thus, they often struggle to scale effectively, particularly in scenarios with large and complex datasets. To mitigate these challenges, many methods use heuristic approaches [3,8] and consider data of limited size, particularly on the number of features. This restricted scope excludes many potential application areas, such as biological and large-scale complex problems.

The differentiable pattern mining framework, as introduced in recent works [9,21], represents a significant advancement in leveraging neural network ar-

chitectures to extract interpretable relations from data. Grounded in neuro-symbolic learning principles, it harnesses the computational power of neural networks to learn fully interpretable patterns within a constrained neural architecture. In the seminal paper by Fischer et al. [9], a novel binarized autoencoder is proposed to uncover human-interpretable sets of conjunctive patterns using gradient-based optimization. This model projects input data into an interpretable latent space, striving to faithfully reconstruct the data from patterns encoded in this space. Interpretability is achieved through the use of binary weights and activations during the forward pass, while scalability is ensured through efficient continuous optimization during backpropagation. Building upon this foundation, Walter et al. [21] extend the framework to learn patterns that can effectively differentiate between classes. Their approach combines a binary autoencoder with a classifier attached to the hidden layer, allowing for joint optimization of reconstruction and classification tasks. This integrated model enhances the interpretability of learned patterns while enabling effective classification of data instances based on these patterns.

While the neural autoencoder with class recovery shows promise, particularly on high-dimensional data, its performance tends to degrade as noise magnitude and the number of underlying classes and features in the data increase. In such scenarios, patterns may become redundant, leaving a substantial portion of the data uncovered. To address this limitation, we propose integrating a novel regularization technique into the differentiable binary model, aimed at promoting the extraction of patterns that provide better coverage of the data while enforcing pattern diversity. Our experiments demonstrate that the orthogonality regularization term in the loss function yields significant improvements in pattern extraction. Additionally, we introduce an innovative pattern decoding strategy that utilizes non-negative matrix factorization (NMF), extending beyond conventional thresholding methods prevalent in existing approaches. This robust and original decoding strategy adapts well to diverse datasets and enhances the overall performance of the model.

The experiments show the scalability of the proposed DiffVersify method concerning the number of features, classes and noise levels, which are pivotal factors in real-world pattern mining scenarios. The evaluation of the effectiveness of the approach on synthetic datasets shows its ability to improve the detection of ground truth patterns with the increase, compared to the baselines, of their similarity with the extracted patterns. Additionally, DiffVersify demonstrates superior performance in terms of the ROC-AUC metric across four real-world datasets. Recognizing that the assessment of pattern collections associated with classes requires more than just supervised classification measures, we introduce novel evaluation metrics to better characterize the appropriateness of discovered patterns relative to the data, with a particular focus on pattern coverage. Our results reveal the effectiveness of DiffVersify on this aspect.

Section 2 reviews related literature. Section 3 introduces notations and concepts while Section 4 outlines the approach; Section 5 presents the experiments, and Section 6 resumes the contributions and discusses their limitations.

2 Related Work

The problem of pattern set and association rule mining was introduced as a method for identifying local structures within data [1]. Rule-based classification, as studied in [5,11,15,19], aims to derive interpretable classification conjunctive rules. Despite featuring interpretability, these methods predominantly prioritize prediction over description, leading to a loss of important contextual details. Furthermore, their reliance on combinatorial optimization techniques hampers their scalability, particularly when applied to high-dimensional datasets. Neuro-symbolic classification [7,14,22] offers a solution to these computational limitations. These methods devise neural architectures that, following training, allow for the extraction of symbolic classification rules. Despite their focus on optimization, these approaches share similarities with traditional rule-based classifiers in their emphasis on classification accuracy rather than descriptive rule discovery. Association discovery research domain experienced a period of robust activity characterized by a plethora of studies, yielding significant insights. Following this first collection of work, the field experienced a resurgence with the introduction of a pioneering neural approach [9], revitalizing research efforts and bringing renewed attention to the domain. In [9], Fischer and Vreeken propose a novel approach, BiNaps, for discovering high-quality and noise-robust pattern sets. Unlike existing methods limited by combinatorial search, BiNaps employs a gradient-based optimization strategy, bridging the discrete search space and continuous optimization. This approach involves a neural autoencoder with binary activations and binarized weights, termed BiNaps, which directly represent conjunctive patterns. By optimizing a data-sparsity aware reconstruction loss, the authors achieve effective pattern discovery, demonstrating scalability to real-world datasets such as supermarket transactions and biological datasets. The patterns are effectively decoded using a thresholded binarisation of the weight matrix of the model after convergence.

In [21], the authors build on BiNaps to propose DiffNaps, a novel binary neural network architecture that builds class-specific patterns. Similarly to BiNaps, DiffNaps also uses a binary autoencoder but combined to a separate classification head. The model is learnt by jointly optimizing reconstruction and classification. Succinct class-specific patterns are promoted thanks to elastic-net regularizers.

Table 1 details the composition of the respective losses and pattern decoding strategies of the main state-of-the-art approaches of the recent literature. While numerous attempts have proposed to take into account various elements of constraints for pattern discovery, our approach DiffVersify proposes two novel contributions. First, we explicitly promote diversity over the resulting pattern set with a dedicated differentiable loss. Second, to enable the discovery of more specific patterns, we propose a novel pattern decoding strategy using latent variable model inference using Non-Negative Matrix factorization for pattern extraction.

Table 1. Comparison of the losses and pattern decoding of baselines and DiffVersify.

		RL-NET [7]	RRL [22]	R2N [14]	BiNaps [9]	DiffNaps [21]	DiffVersify
Losses	Reconstruction	✓	✗	✗	✓	✓	✓
	Classification	✗	✓	✓	✗	✓	✓
	$\mathcal{L}_1$ regularization	✗	✗	✓	✗	✓	✓
	$\mathcal{L}_2$ regularization	✓	✓	✗	✗	✓	✓
	Coverage regularization	✗	✗	✗	✗	✗	✓
Decoding	Threshold	✗	✗	✗	✓	✓	✓
	NMF	✗	✗	✗	✗	✗	✓

3 Preliminaries

We assume a supervised input dataset $\mathcal{D} : (\mathbf{X}, \mathbf{Y})$ with $\mathbf{X} \in \{0,1\}^{n \times m}$ composed with n samples and m features, and $\mathbf{Y} \in [0,1]^{n \times k}$, the probability for each sample to be assigned to one of the class labels $K = \{1, \ldots, k\}$. Our purpose consists in finding a set of patterns P, where each pattern $p \in P$ is a set of feature indices $p \subset \{1, 2, \ldots, m\}$ representing feature co-occurrences. To find such sets of patterns, it has been recently proposed to learn a binarized autoencoder type of neural network, where $\mathbf{W} \in \mathbb{R}^{m,h}$ is its weight matrix with h hidden dimensions. We denote by $\mathbf{W}_i$, the i-th row of $\mathbf{W}$. $\mathbf{W}^d$ indicate the binarized version of $\mathbf{W}$. We also consider b for a bias, and b^d for its discretized value. For a given binary database, our aim is to find a diverse set of patterns P that describes the data. One interpretation of this claim consists in defining a set of patterns as correct if it can marginally reconstruct the database.

4 Differentiable pattern mining with coverage regularization

Pattern mining has been recently tackled using autoencoders, minimizing reconstruction loss with additional class prediction, facilitating robust pattern discovery. As a first contribution, we introduce a diversity objective to minimize collapsing among neurons of the encoder layer during training. Secondly, we propose a novel decoding process after training, promoting the creation of longer patterns with respect to solely thresholding using Non-Negative Matrix Factorization (NMF).

4.1 Neural model for pattern mining

For various data, autoencoders have proven to be a successful approach for capturing the main regularities in the data by minimizing reconstruction loss. An autoencoder is a neural network consisting of task-specific encoding layers that end in an embedding layer, and a symmetric decoder to reconstruct the input from the embedding layer. The embedding layer is usually small compared to

the input layer, imposing an information bottleneck and forcing the network to learn relevant and shared structure between inputs.

To support interpretability, a novel type of neural autoencoder as been recently proposed, where weights and activations are discretized in $\{0, 1\}$ during the forward pass. To learn in small noisy steps during backpropagation, for training continuous versions of the weights are used, optimizing reconstruction loss with respect to these continuous weights. The autoencoder consists of one linear hidden layer – a so-called pattern layer – and one linear output layer. For each neuron in the hidden layer, incoming binary weights indicate whether an input item is part of the encoded pattern. For example, a binarized weight $\mathbf{W}^d_{i,j}$ means that input item j is part of the pattern given by hidden neuron i. Thus, each neuron in the hidden layer corresponds to a pattern p, while all neurons together correspond to the pattern set P.

Concretely, one binarized version of the weights is construct for compute the forward pass, and used for reconstruction. To ensure that the hidden neurons correspond to interpretable patterns, the auto-encoder architecture is symmetrical as the weight of the decoding layer is the transpose of the weight of the encoding layer.

4.2 Learning algorithm

The architecture of differentiable pattern recognition usually consists of a binary autoencoder. The encoding and decoding layers of the autoencoder share a set of continuous weights $\mathbf{W}$. The forward pass uses a binarized version of this weight matrix $\mathbf{W}^d$ following [9]. Each hidden neuron j represents a pattern, and a feature i is part of the pattern corresponding to neuron j if $\mathbf{W}^d_{i,j} = 1$. The decoding layer performs the transposed linear transformation of the encoding layer which enforces the patterns formed during optimization to describe the data. In recent work, a classifier has been added to the pattern layer with continuous weights $\mathbf{W}^c$ to act as an additional regularizer. This classifier is linear and hence interpretable.

The overall objective function consists of a series of terms for the autoencoder reconstruction, the classification error, and various regularization terms.

Reconstruction Loss. First, the autoencoder reconstruction loss from the input points is defined with a weighted XOR function as proposed in [9]. As binary data tends to be sparse and dominated toward zeros, a sparsity-aware reconstruction loss weighs the importance of reconstructing a 1 proportional to the sparsity of the data.

$$\mathcal{L}_e(\mathbf{X}_i, \widehat{\mathbf{X}_i}) = \sum_{j=1}^{m}((1 - \mathbf{X}_{i,j})\alpha + \mathbf{X}_{i,j}(1 - \alpha))|\widehat{\mathbf{X}_{i,j}} - \mathbf{X}_{i,j}|, \tag{1}$$

with $\alpha = \frac{\#1s}{\#1s+\#0s}$ the sparsity of $\mathbf{X}$ and $\widehat{\mathbf{X}}$ the reconstruction of $\mathbf{X}$ by the autoencoder.

Classification Loss. Second, to optimize the classifier, the cross-entropy loss is naturally optimized between the predicted logits $\widehat{\mathbf{Y}}$ and the true label $\mathbf{Y}$:

$$\mathcal{L}_c(\mathbf{Y}_i, \widehat{\mathbf{Y}_i}) = -\sum_{\ell=1}^{k} \mathbf{Y}_{i,\ell} \log(\widehat{\mathbf{Y}}_{i,\ell}) \tag{2}$$

***L*$_2$-regularizer.** Next, to promote parsimonious patterns, the L_2-regularizer is leveraged to penalize long patterns, i.e., rows with many 1s. The function $r_s(\mathbf{W})$ is defined as:

$$r_s(\mathbf{W}) = \sum_{i=1}^{m} \left(\sum_{j=1}^{h} \mathbf{W}_{i,j} \right)^2 \tag{3}$$

This function computes the squared sum of each row of the weight matrix $\mathbf{W}$. This loss penalizes a pattern as a whole as it defines a quadratic cost on the length of the pattern. Hence, the regularizer is promoting shorter patterns discovery.

W-shaped regularizer. To further force the weights towards a binary solution, a W-shaped regularizer is defined. The function $r_b(\mathbf{W})$ is defined as:

$$r_b(\mathbf{W}) = \min_i \{r(\mathbf{W}_i), r(\mathbf{W}_i - 1)\}, \tag{4}$$

where $r(\mathbf{W}_i)$ is defined as $r(\mathbf{W}_i) = \kappa\|\mathbf{W}_i\|_1 + \lambda\|\mathbf{W}_i\|_2^2$. Here, κ and λ are hyperparameters specifying the trade-off between the L_1 and L_2 regularization penalties. This regularizer takes the classic form of an elastic-net, where the κ and λ hyperparameters respectively specify the trade-off between the ridge and lasso penalty.

Coverage regularization. Finally, to enforce diversity and data-coverage among the patterns in the model's representations, we introduce a orthogonality component into the loss function. By including the orthogonality constraint in the loss function, the model is encouraged to learn diverse and independent features, which can lead to improved generalization performance. The orthogonality constraint is defined through a cosine similarity between each pair of the neurons, which correspond to a line of $\mathbf{W}$. By encouraging orthogonality, the loss function helps prevent the model from collapsing to specific features and encourages it to learn more informative representations. Formally, the loss is defined as follows:

$$\mathcal{L}_{\text{cov}}(\mathbf{W}) = \frac{1}{m(m-1)} \sum_{i \neq j} \left(\frac{\mathbf{W}_i \cdot \mathbf{W}_j}{\|\mathbf{W}_i\| \|\mathbf{W}_j\|} \right). \tag{5}$$

This orthogonality component is combined with other regularization terms to form the complete loss function. As a result, given the parameters of the

network $\{\mathbf{W}, \mathbf{W}^c\}$, the loss function is given by:

$$\mathcal{L}(\mathcal{D}, \mathbf{W}, \mathbf{W}^c\}) = \sum_{i=1}^{n} \Big[\mathcal{L}_e(\mathbf{X}_i, \widehat{\mathbf{X}}_i) + \lambda_c \mathcal{L}_c(\mathbf{Y}_i, \widehat{\mathbf{Y}}_i)\Big] + r_s(\mathbf{W}) + r_b(\mathbf{W}) + r_b(\mathbf{W}^c) + \mathcal{L}_{\text{COV}}(\mathbf{W}), \tag{6}$$

where λ_c is a parameter that weighs the classification loss.

4.3 Pattern decoding from latent representation

To extract differential patterns at convergence, $\mathbf{W}$ and $\mathbf{W}^c$ are classically thresholded with τ_e and τ_c, respectively. As described above, a pattern p_j is given by the index set of all i's such that $\mathbf{W}^d_{i,j} = 1$. However, one limitation can be mentioned: the decoding process does not consider the creation of long patterns, resulting from the coverage loss. These longer patterns capture intricate dependencies and interactions between features, possibly offering a deeper understanding of the underlying data structure. Unfortunately, the decoding mechanism may overlook these longer patterns, potentially leading to a loss of information during the reconstruction phase. As a result, the reconstructed data may lack the finer details captured by these longer patterns, hindering the fidelity of the reconstructed dataset. This limitation underscores the need for a decoding strategy that can effectively incorporates the information encoded in longer patterns. So, we propose to improve the pattern decoding process using Non-Negative Matrix Factorization (NMF) over $\mathbf{M}$ define as

$$\mathbf{M}_{i,j} = \frac{\sum_{\ell=0}^{m} \mathbf{X}_{i,\ell}.\mathbf{W}^d_{\ell,j}}{\sum_{\ell=0}^{m} \mathbf{W}^d_{\ell,j}} \tag{7}$$

This way, one can improve the quality and accuracy of the reconstructed patterns.

Non-negative Matrix Factorization is a popular technique of dimensionality reduction that has been explored in numerous applications, like topic modelling and recommendation systems [2]. Given a non-negative matrix $\mathbf{M}$ with dimensions $m \times h$, NMF seeks to factorize this matrix into two non-negative matrices $\mathbf{U}$ and $\mathbf{V}$, such that $\mathbf{M} \approx \mathbf{UV}$. Here, $\mathbf{U}$ represents a basis matrix with dimensions $m \times g$, where g is typically chosen to be smaller than m and h, and $\mathbf{V}$ denotes a coefficient matrix with dimensions $g \times h$. The factorization is constrained to be non-negative, meaning that all elements of $\mathbf{U}$ and $\mathbf{V}$ are non-negative. The resulting factorization aims to represent $\mathbf{M}$ as a linear combination of a reduced set of basis vectors from $\mathbf{U}$, weighted by the coefficients in $\mathbf{V}$. The objective loss function of NMF is defined as the Frobenius norm of the difference between the original matrix and the reconstructed matrix:

$$\mathcal{L}_{\text{NMF}}(\mathbf{M}, \widehat{\mathbf{M}}) = \|\mathbf{M} - \widehat{\mathbf{M}}\|_F^2 = \|\mathbf{M} - \mathbf{UV}\|_F^2. \tag{8}$$

As for topic modeling in natural language processing [23,20], we build new and longer patterns by aggregating the top-p patterns defined as

$$\text{top-p} = \text{argmax}_{j=1\ldots p} \mathbf{V}_{d,j}$$

that corresponds to the most co-occurred patterns for each latent dimension $d \in [1, g]$ of the matrix V. This approach aims at regrouping the patterns that frequently co-occurs in the data. This composition is complementary to the proposed coverage loss which tend to create shorter and orthogonal patterns.

5 Experiments

In the following experiments, we address the following questions. First, we assess how our proposed model performs with respect to the current state-of-the-art methods in terms of scalability, robustness, and overall effectiveness, across both real-world and synthetic datasets. Second, we assess how the diversity regularizer allows to discover a larger variety of discriminative patterns. Third, we question how the proposed NMF decoding build more specific patterns beyond generalist ones in real and synthetic data[4].

5.1 Metrics

We use a set of metrics to assess the pertinence of the discovered patterns. Indeed, the sheer volume and complexity of the patterns generated makes it challenging to identify the most relevant and informative ones. So, several metrics can help to assess the quality, novelty, and usefulness of patterns, and to identify those that are most likely to be of interest to domain experts or end-users. Let P be the set of patterns found and $\mathbf{z}_p$ is the binary vector denoting the assignment of the dataset point to the support of pattern p. We use the following measures to describe the collection of patterns:

- Cover: It computes the proportion of the dataset samples covered by at least one pattern: $\frac{||\bigvee_{p \in P} \mathbf{z}_p||}{n}$.
- Purity: It measures the purity of a pattern with respect to $\mathbf{y}$: $\frac{1}{|P|} \sum_{p \in P} \frac{\max_\ell ||\mathbf{y}_\ell \bigwedge \mathbf{z}_p||}{||\mathbf{z}_p||}$

Then, we use a set of measures to evaluate the collection of patterns as a prediction model:

- Weighted-F1: For each sample, we take the set of patterns that support it. Among those patterns, we select one with the highest purity and associate this class as the predicted label for the sample. In the case where no pattern is supporting a sample, the majority class is associated to it. Weighted-F1 score calculates the F1 score for each class and then computes a weighted average based on the number of samples in each class.
- ROC-AUC based on Purity and Cover: Since ground truth labels are not available for real-world data, we evaluate the collection of patterns as presented in [21], by using the area under the curve of the percentage of data covered by patterns (Cover measure) once patterns are sorted according to

[4] Code and data are available: https://chataingt.github.io/DiffVersify/.

their PURITY that is proportional to the probability of predicting the target class). This evaluation can be interpreted as a trade-off between sensitivity, i.e. the proportion of the dataset covered, and specificity, i.e. the pattern's relevance to a particular class. To eliminate spurious patterns, we only consider those with a predictive probability of $\frac{1}{k} + 0.1$ or higher, indicating a slightly greater likelihood than chance.

- SOFT-F1: When the ground truth patterns are accessible, as usually in synthetic datasets, we utilizes the Jaccard distance instead of strict equality for calculating recall and precision as it prevents an excessive penalty for methods that only partially recover individual patterns [12]. The SOFT-F1 score is defined as the harmonic mean of soft precision and soft recall defined by:
  soft precision$(P_d, P_g) = \frac{1}{|P_d|} \sum_{p_d \in P_d} \max_{p_g \in P_g} \frac{|p_d \cap p_g|}{|p_d \cup p_g|}$
  soft recall$(P_d, P_g) = \frac{1}{|P_g|} \sum_{p_g \in P_g} \max_{p_d \in P_d} \frac{|p_d \cap p_g|}{|p_d \cup p_g|}$
  where soft recall and soft precision are computed using the Jaccard distance between the recovered and ground truth patterns. The soft F1 score allows to take into accounts partial matches between recovered and ground truth patterns.

We also consider other description measures of pattern collection:

- # PATTERNS: The number of patterns in P.
- AVG. SUPP.: The average support of the patterns in P: $\frac{\sum_{p \in P} ||\mathbf{z}_p||}{\#\ \text{PATTERNS}}$.

5.2 Baselines

We evaluate our model against the seminal proposal of BINAPS and DIFFNAPS, its improvement in class-specific pattern set mining. By transitivity, we challenge the current state-of-the-art methodologies including decision trees, significant pattern mining [18], MDL-based label-descriptive approaches [12], classification rule learning [19], neuro-symbolic classification rule learning [22], top-k subgroup discovery [16], difference description [4], falling rule lists [17], optimal sparse decision trees [11], and class-specific BMF [13] reported by DIFFNAPS [21]. Indeed, DIFFNAPS has superior performance than these baselines. Throughout all experiments, we utilize the replication package of DIFFNAPS to establish parameters for consistency across the subsequent experiments.

5.3 Experiments on real-world benchmarks

First, we evaluate DIFFVERSIFY on four biology-related benchmarks with the variant DIFFVERSIFY-ABL to do an ablation study over the use of the non-negative factorization. The impact of the diversity regularizer is evaluated through the comparison with DIFFNAPS.

Datasets. We consider a phenotypical CARDIO dataset[5], a DISEASE diagnosis dataset[6] and two high-dimensional binarized gene expression datasets for

[5] `https://www.kaggle.com/datasets/sulianova/cardiovasculardisease-dataset.`
[6] `https://www.kaggle.com/datasets/itachi9604/diseasesymptom-description-dataset.`

breast cancer, BRCA-N and BRCA-S, both derived from The Cancer Genome Atlas (TCGA)[7]. The number of descriptive features are respectively 45, 131, 1976 and 1976. The number of individuals are respectively, 68k, 5k, 222 and 187. The number of classes are respectively 2, 41, 2 and 4. We use the hyper-parameters reported in DIFFNAPS and BINAPS, which were optimized on these dataset, and we use cross-validation to define the ones for DIFFVERSIFY. In particular, the rank g of NMF decoding determined for each real-world dataset is as follows: CARDIO: 10, DISEASE: 15, BRCA-N: 100 and BRCA-S: 500.

Table 2. Comparison of performance metrics across four real-world datasets. Average values and standard-deviations are reported over 5 runs of the methods.

	Measures	Methods	Datasets			
			BRCA-N	BRCA-S	CARDIO	DISEASE
Model evaluation	ROC AUC	BINAPS	0±0	0±0	0.06±0.15	0.76±0.01
		DIFFNAPS	0.90±0.05	0.79±0.04	0.34±0.05	0.84±0.0
		DIFFVERSIFY-ABL	0.92±0.00	0.89±0.03	0.54±0.02	0.86±0.01
		DIFFVERSIFY	**0.95±0.00**	**0.93±0.03**	**0.55±0.18**	**0.90±0.01**
	WEIGHTED-F1	BINAPS	0±0	0±0	0.34±0.0	0.76±0.03
		DIFFNAPS	0.55±0.21	0.18±0.19	**0.71±0.01**	0.88±0.04
		DIFFVERSIFY-ABL	0.63±0.28	0.20±0.07	0.68±0.02	0.98±0.00
		DIFFVERSIFY	**0.79±0.25**	**0.38±0.14**	0.69±0.02	**1.00±0.01**
Model description	COVER	BINAPS	0±0	0±0	0.87±0.07	0.79±0.03
		DIFFNAPS	**1.00±0.00**	**1.00±0.00**	0.66±0.17	0.99±0.01
		DIFFVERSIFY-ABL	**1.00±0.00**	**1.00±0.00**	**0.98±0.02**	**1.00±0.00**
		DIFFVERSIFY	**1.00±0.00**	**1.00±0.00**	**0.98±0.02**	**1.00±0.00**
	PURITY	BINAPS	0±0	0±0	0.54±0.05	**0.98±0.01**
		DIFFNAPS	**0.84±0.03**	**0.37±0.05**	**0.77±0.04**	0.13±0.00
		DIFFVERSIFY-ABL	0.59±0.01	0.35±0.02	0.73±0.03	0.10±0.00
		DIFFVERSIFY	0.61±0.02	**0.37±0.02**	**0.77±0.02**	0.22±0.00
Other measures	# PATTERNS	BINAPS	0±0	0±0	5.80±1.92	124.60±2.07
		DIFFNAPS	182.60±40.83	939.20±336.21	10.06±1.14	3693.69±206.63
		DIFFVERSIFY-ABL	2674.80±1088.92	9630.00±1398.29	8.20±1.64	2626.40±59.58
		DIFFVERSIFY	2874.80±1088.93	11630.00±1398.29	22.2±1.64	3241.40±59.58
	AVG. SUPP.	BINAPS	0±0	0±0	49849.89±8703.88	95.22±3.61
		DIFFNAPS	33.02±0.58	26.08±1.79	9731.81±4711.73	275.12±7.21
		DIFFVERSIFY-ABL	31.05±0.58	26.26±1.79	18231.42±2514.64	273.41±5.07
		DIFFVERSIFY	29.83±0.60	24.29±1.61	14938.36±4172.42	263.72±4.86

Results. Table 2 reports the measure values obtained by the different methods on the 4 datasets. First, notice that BINAPS returns 0 patterns on both BRCA datasets and therefore the measures can not be evaluated. For all dataset, we can observe that DIFFVERSIFY's patterns exhibit perfect COVER, indicating a complete representation of the data. Notice that there is a large number of patterns on BRCA datasets due to their high number of features compared to their number of data points.

[7] The BRCA datasets were derived from data made available by the TCGA Research Network.

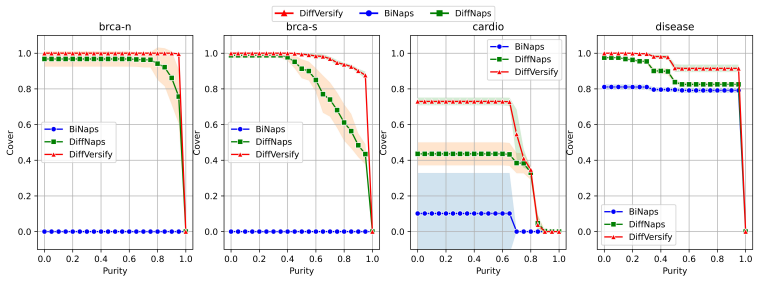

Fig. 1. ROC curve on four biology-related benchmarks, BRCA-N, BRCA-S, CARDIO and DISEASEover 5 runs.

The ROC-AUC shows that DIFFVERSIFY's patterns consistently provide superior COVER with better PURITY, suggesting that they are more effective in describing the classes. Fig. 1 shows the ROC curves. For BRCA-S and BRCA-N datasets, we can observe the critical role of regularization for this metric (see the increase compared to DIFFNAPS). This is particularly pertinent with datasets that exhibit a disparity between the number of rows and columns. When applied to datasets such as CARDIO and DISEASE, both DIFFVERSIFY and DIFFNAPS demonstrate comparable performance levels. However, as the complexity of the dataset increases, DIFFVERSIFY seems to generate better patterns. Indeed, these patterns offer better coverage while maintaining good PURITY, thereby underscoring the potential efficacy of DIFFVERSIFY.

The WEIGHTED-F1 score, in conjunction with the perfect coverage, underscores the ability of DIFFVERSIFY's pattern set to discriminate between classes, even when the pattern set covers all the samples of the dataset. This indicates that DIFFVERSIFY not only provides thorough coverage but also maintains a high discriminating capability. The ablation study, where the NMF step is excluded, demonstrates that although DIFFVERSIFY-ABL may outperform DIFFVERSIFY on ROC-AUC, as on BRCA-N, DIFFVERSIFY systematically benefits from this post-processing in all other performance measures.

However, it is worth noting that DIFFVERSIFY identified a substantially higher number of patterns (# PATTERNS) in three of the datasets. This can be attributed to the diversity constraint, which facilitates the generation of more general patterns, and the subsequent NMF decoding that refines these patterns into more precise ones.

This limitation can be addressed through a straightforward yet efficient procedure: sorting the patterns based on their COVER value and selecting patterns until achieving a coverage of 1. Fig. 2 illustrates the performance metrics achieved with an increasing set of selected patterns. The results indicate that this post-processing leads to a more compact and effective pattern set. In BRCA-N, merely one hundred patterns yield satisfactory performance, aligning the # PATTERNS value with the minimum observed in other methods. On BRCA-S, a trade-off between WEIGHTED-F1 and ROC-AUC can be achieved applying an

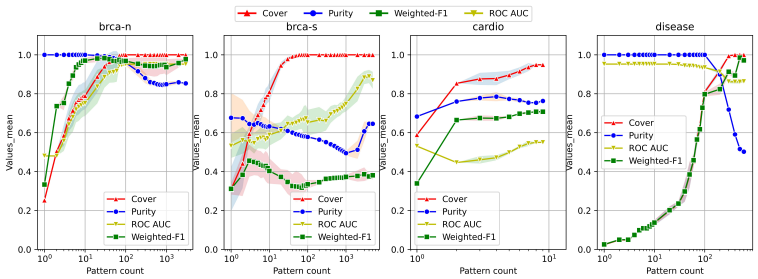

Fig. 2. Evolution of the performance metrics of DiffVersify over subsets of patterns defined by increasing Cover values across BRCA-N, BRCA-S, CARDIO and DISEASEover 5 runs.

elbow method on Cover measure. However, in CARDIO, where the pattern set is already small, this post-processing step is deemed unnecessary.

Finally Table 3 reports qualitative results. Considering in detail top patterns with respect to Cover found per class on CARDIO by all three methods (BiNaps, DiffNaps and DiffVersify), Table 3 reveals that DiffVersify consistently identifies at least the same set of patterns as the baseline DiffNaps. Remarkably, DiffVersify outperforms DiffNaps by uncovering additional patterns, characterized by high coverage and purity scores, that the latter fails to detect.

5.4 Experiments on synthetic data

To enhance the understanding and comparison of the different methods, we use synthetic data to readily access ground truth patterns, providing a controlled environment for evaluating the properties of the considered approaches.

Dataset generation. For the data generation process, we use the publicly available DiffNaps replication package. Within each class, ten patterns are randomly sampled across features, with lengths drawn from a uniform distribution ($U(5, 15)$). Also, 20 common patterns are sampled, with lengths drawn from $U(0.01 \times m, 0.025 \times m)$ to maintain data density. Each class comprises an equal number of samples, each containing two common and three class-specific patterns randomly embedded. We introduce additive and destructive noise by flipping ten 0s to 1s and flipping 1s affected by a pattern to 0s with a 2.5% probability, respectively. Class labels are assigned to satisfy $\frac{(\mathbf{z}_p^t \mathbf{Y})_k}{n} = 0.9$. Means and std of measures across four independently generated datasets are reported. We set the rank g for NMF decoding equal to the number of ground truth patterns.

Scalability in m. One significant challenge in existing pattern-set mining approaches is handling high-dimensional data. We thus vary the number of features m within $\{10^2, 5\times10^2, 10^3, 5\times10^3, 10^4, 1.5\times10^4, 2\times10^4, 2.5\times10^4, 5\times10^4, 10^5\}$. We set the number of classes to $k = 2$ and the number of rows to $n = 10^4$. To mitigate pattern overlap in low-dimensional data ($m < 10^3$), we sample 5 patterns per class without sharing.

Table 3. Analysis of the top 6 patterns in terms of Cover by class on cardio and thus for the three methods: BiNaps, DiffNaps and DiffVersify. A pattern is mentioned as Unique if only one of the methods discovered it.

c	Method	Unique	Cover	Cover[c]	Purity	Features
Heart attack	DiffVersify	✓	0.28	0.44	0.77	ap_lo_High_2
	DiffNaps		0.26	0.45	0.84	ap_hi_High_2
	DiffVersify		0.26	0.45	0.84	ap_hi_High_2
	DiffNaps		0.20	0.35	0.84	ap_hi_High_2, ap_lo_High_2
	DiffVersify		0.20	0.35	0.84	ap_hi_High_2, ap_lo_High_2
	DiffVersify	✓	0.16	0.27	0.83	ap_hi_High_2, cholesterol_normal
	DiffNaps		0.14	0.20	0.70	age_(60.0, 64.0)
	DiffVersify		0.14	0.20	0.70	age_(60.0, 64.0)
	DiffNaps		0.11	0.18	0.76	cholesterol_way_above
	DiffVersify		0.11	0.18	0.76	cholesterol_way_above
Healthy	DiffVersify	✓	0.71	0.86	0.62	ap_lo_Normal_Elevated
	DiffNaps		0.59	0.80	0.68	ap_hi_Normal
	DiffVersify		0.59	0.80	0.68	ap_hi_Normal
	DiffVersify	✓	0.57	0.74	0.66	ap_lo_Normal_Elevated, cholesterol_normal
	DiffNaps		0.57	0.77	0.69	ap_hi_Normal. ap_lo_Normal_Elevated
	DiffVersify		0.57	0.77	0.69	ap_hi_Normal, ap_lo_Normal_Elevated
	BiNaps	✓	0.23	0.33	0.73	gender_women, ap_hi_Normal ...
	DiffNaps		0.18	0.24	0.69	age_(29.0, 45.0)
	DiffVersify		0.18	0.24	0.69	age_(29.0, 45.0)

Multi-classes. We assess the methods' capability to classify data as the number of distinct classes increases. The number of classes k ranges from 2 to 50, with 4×10^3 samples generated per class and $m = 5 \times 10^3$ features.

Robustness to additive noise. We evaluate the robustness of our model to additive noise, by simulating scenarios in which data may be corrupted or perturbed. Setting $k = 2$, $m = 5 \times 10^3$, and $n = 10^3$, we introduce additive noise by varying the number of randomly added 1s per row from 0 to 100.

Robustness to destructive noise. The robustness of the model against destructive noise, a significant challenge in extracting meaningful patterns, is evaluated by varying the probability of flipping 1s to 0s from 0% to 60%.

Results. The results are shown in Fig. 3 and Table 4. In the feature and noise experiments in Table 4, we expect to identify 20 ground truth patterns. Remarkably, DiffVersify is the only method that consistently achieves near-perfect coverage, irrespective of the magnitude of the noise or the value of dimensionality. Both DiffVersify and DiffNaps discover class-specific patterns with an average purity surpassing 0.8 across all experiments. Notably, DiffVersify shows higher similarity to the ground truth patterns relative to its baselines, a phenomenon attributable to the NMF decoding process. It is worth mentioning that the addition of obtained patterns to the existing ones invariably introduces similarity among patterns. Furthermore, both DiffVersify and DiffNaps manage to identify the majority of the ground-truth, with an average soft-F1 score exceeding 0.70. The exception to this observation is in the experiment involving the

Table 4. Performance comparison on synthetic datasets.

	Measures	Method	# Features	# Classes	Add. noise	Dest. noise
Model evaluation	Soft-F1	BiNaps	0.24±0.3	0.52±0.21	0.09±0.08	0.07±0.03
		DiffNaps	**0.89±0.12**	0.59±0.09	0.65±0.05	0.5±0.17
		DiffVersify	**0.81±0.23**	**0.66±0.09**	**0.73±0.07**	**0.68±0.13**
	Weighted-F1	BiNaps	0.63±0.15	0.11±0.16	0.61±0.04	0.57±0.03
		DiffNaps	**0.88±0.03**	0.65±0.12	0.76±0.01	0.56±0.18
		DiffVersify	**0.89±0.02**	**0.75±0.11**	**0.84±0.06**	**0.62±0.2**
Model description	Cover	BiNaps	**1.0±0.01**	**1.0±0.0**	**1.0±0.0**	**0.99±0.01**
		DiffNaps	**0.95±0.06**	0.79±0.2	0.73±0.03	0.38±0.28
		DiffVersify	**0.99±0.02**	**0.85±0.17**	0.84±0.9	0.48±0.35
	Purity	BiNaps	0.71±0.08	0.53±0.05	0.73±0.05	0.7±0.02
		DiffNaps	**0.9±0.01**	**0.83±0.08**	**0.9±0.02**	**0.88±0.06**
		DiffVersify	**0.87±0.04**	**0.84±0.08**	**0.9±0.02**	**0.9±0.06**
Other measures	# Patterns	BiNaps	369.51±259.53	–	323.35±93.86	203.02±22.48
		DiffNaps	17.8±4.87	–	14.13±2.05	9.73±4.95
		DiffVersify	41.36±29.23	–	16.85±2	13.47±6.7
	Avg. Supp.	BiNaps	1896.77±1136.14	1206.58±679.08	119.8±32.5	118.38±8.06
		DiffNaps	2667.88±1106.17	210.69±88.06	170±28.1	106.17±64.27
		DiffVersify	2664.29±1294.1	188.88±38.18	188.51±24.31	102.27±75.84

number of classes, where the complexity of the classes diminished performance to a level akin to the baseline, BiNaps. In terms of weighted-F1 results, DiffVersify and DiffNaps show comparable performance, although DiffVersify exhibits superior average results. As the number of class is varying, the number of groundtruth patterns is varying accordingly. As a consequence, we do not compute the number of patterns for these specific experimental settings in the table. In Fig. 3, DiffVersify exhibits better robustness to both additive and destructive noises. Furthermore, DiffVersify demonstrates scalability with respect to the number of features, particularly in high-dimensional settings where it outperforms DiffNaps in terms of robustness.

The results suggest that our proposed method is able to discover more diverse and discriminative patterns compared to the baseline methods, while maintaining high coverage and purity. The use of diversity regularization and NMF decoding in DiffVersify allows for the discovery of longer and more specific patterns, which lead to improved generalization performance. Further work could be done to improve the proposed approach in several ways. One potential direction is to explore other decoding strategies beyond NMF, such as using more advanced matrix factorization techniques or incorporating domain-specific knowledge into the decoding process. Another direction is to investigate the use of other regularization techniques, such as group-sparsity regularizers, to further encourage diversity and interpretability in the learned patterns. One limitation of our proposed approach is that it relies on the assumption that the data can be well-represented by a set of binary patterns. However, in some cases, the data may contain more complex relationships that cannot be captured by binary patterns

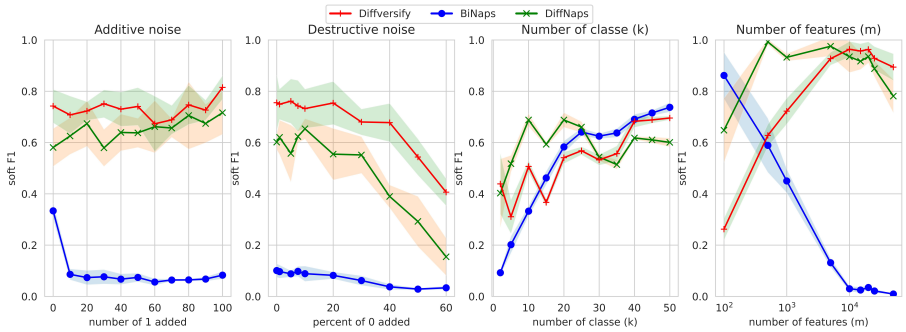

Fig. 3. F1 scores obtained on the synthetic datasets by varying (from left to right) the additive noise level, the destructive noise level, k and m.

alone. In such cases, it may be necessary to extend the approach to allow for more complex pattern representations, such as real-valued or continuous patterns.

6 Conclusion

We introduced a novel differentiable binary model for pattern mining that incorporates a regularization loss emphasizing pattern coverage and a pattern decoding strategy using non-negative matrix factorization (NMF). Our approach demonstrates superior performance in terms of ROC-AUC on four real-world biology-related datasets and improves pattern detection by increasing similarity measure to ground truth patterns on synthetic data. Through extensive evaluations, we show the appropriateness of discovered patterns relative to the data, focusing on pattern coverage, indicating the efficacy of our approach in handling challenging scenarios with high noise levels and multiple classes. One possible future direction is to search for alternative techniques for pattern decoding from differentiable model. This could help to better capture complex patterns and improve the overall accuracy of our approach. Another direction is to incorporate additional regularizers such as those proposed in neuro-symbolic approaches.

Acknowledgement

This work benefited from state aid managed by the National Research Agency under France 2030 with the reference "ANR-22-PEAE-0008".

References

1. Agrawal, R., Imielinski, T., Swami, A.N.: Mining association rules between sets of items in large databases. In: SIGMOD. pp. 207–216. ACM Press (1993)
2. Berman, A., Plemmons, R.J.: Nonnegative matrices in the mathematical sciences. In: Classics in Applied Mathematics (1979),

3. Bosc, G., Boulicaut, J., Raïssi, C., Kaytoue, M.: Anytime discovery of a diverse set of patterns with monte carlo tree search. DAMI **32**(3), 604–650 (2018)
4. Budhathoki, K., Vreeken, J.: The difference and the norm: Characterising similarities and differences between databases. In: Mach (2015)
5. Dash, S., Günlük, O., Wei, D.: Boolean decision rules via column generation. In: NeurIPS. pp. 4660–4670 (2018),
6. De Bie, T.: Maximum entropy models and subjective interestingness: an application to tiles in binary databases. Data Min. Knowl. Discov. **23**(3), 407–446 (2011)
7. Dierckx, L., Veroneze, R., Nijssen, S.: Rl-net: Interpretable rule learning with neural networks. In: PAKDD. pp. 95–107 (2023)
8. Dzyuba, V., van Leeuwen, M., Raedt, L.D.: Flexible constrained sampling with guarantees for pattern mining. Data Min. Knowl. Discov. **31**(5), 1266–1293 (2017)
9. Fischer, J., Vreeken, J.: Differentiable pattern set mining. In: SIGKDD. pp. 383–392. ACM (2021)
10. Gionis, A., Mannila, H., Mielikäinen, T., Tsaparas, P.: Assessing data mining results via swap randomization. ACM Trans. Knowl. Discov. Data **1**(3), 14 (2007)
11. Hayden, M., Chudi, Z., Reto, A., Ilias, K., Jacques, C. *et al.*: Fast sparse decision tree optimization via reference ensembles. In: AAAI. vol. 36 (2022)
12. Hedderich, M., Fischer, J., Klakow, D., Vreeken, J.: Label-descriptive patterns and their application to characterize classification errors. In: ICML (2022)
13. Hess, S., Morik, K.: C-SALT: mining class-specific alterations in boolean matrix factorization. In: ECML PKDD. vol. 10534, pp. 547–563 (2017)
14. Kusters, R., Kim, Y., Collery, M., Marie, C.d.S., Gupta, S.: Differentiable rule induction with learned relational features. arXiv preprint arXiv:2201.06515 (2022)
15. Lakkaraju, H., Bach, S.H., Leskovec, J.: Interpretable decision sets: A joint framework for description and prediction. In: SIGKDD. pp. 1675–1684 (2016)
16. Lemmerich, F., Becker, M.: pysubgroup. In: Machine Learning and Knowledge Discovery in Databases, vol. 11053, pp. 658–662. springer (2019)
17. Lin, J.J., Zhong, C., Hu, D., Rudin, C., Seltzer, M.I.: Generalized and scalable optimal sparse decision trees. In: ICML (2020)
18. Pellegrina, L., Riondato, M., Vandin, F.: Spumante: Significant pattern mining with unconditional testing. In: SIGKDD. pp. 1528–1538 (2019)
19. Proença, H.M., van Leeuwen, M.: Interpretable multiclass classification by mdl-based rule lists. Information Sciences **512**, 1372–1393 (2020)
20. Shi, T., Kang, K., Choo, J., Reddy, C.K.: Short-text topic modeling via nmf enriched with local word-context correlations. WWW (2018)
21. Walter, N.P., Fischer, J., Vreeken, J.: Finding Interpretable Class-Specific Patterns through Efficient Neural Search. In: AAAI (2024)
22. Wang, Z., Zhang, W., Liu, N., Wang, J.: Scalable rule-based representation learning for interpretable classification. NeurIPS **34**, 30479–30491 (2021)
23. Xu, W., Liu, X., Gong, Y.: Document clustering based on non-negative matrix factorization. ACM SIGIR (2003),
24. Zaki, M.J., Parthasarathy, S., Ogihara, M., Li, W.: New algorithms for fast discovery of association rules. In: SIGKDD. pp. 283–286 (1997)