Learning Exceptional Subgroups by End-to-End Maximizing KL-divergence

Sascha Xu [*1] **Nils Philipp Walter** [*1] **Janis Kalofolias** [1] **Jilles Vreeken** [1]

Abstract

Finding and describing sub-populations that are exceptional in terms of a target property has important applications in many scientific disciplines, from identifying disadvantaged demographic groups in census data to finding conductive molecules within gold nanoparticles. Current approaches to finding such *subgroups* require pre-discretized predictive variables, do not permit non-trivial target distributions, do not scale to large datasets, and struggle to find diverse results. To address these limitations, we propose SYFLOW, an end-to-end optimizable approach in which we leverage normalizing flows to model arbitrary target distributions and introduce a novel neural layer that results in easily interpretable subgroup descriptions. We demonstrate on synthetic data, real-world data, and via a case study, that SYFLOW reliably finds highly exceptional subgroups accompanied by insightful descriptions.

1. Introduction

The majority of modern machine learning focuses on finding global models that perform well on *predictive* tasks such as classification. In this domain, deep neural networks often achieve state-of-the-art performance, at the expense of human-interpretable insight.

Orthogonal to the advances in predictive modeling, many scientific applications require *descriptive* modeling such as finding sub-populations that are somehow *exceptional*, and providing a human-interpretable description for these. Applications of finding such *subgroups* range from identifying disadvantaged demographic groups in census data (Boll & Lagemann, 2019; Ortiz & Cummins, 2011) to learning those combinations of properties that single out materials with desirable properties (Sutton et al., 2020).

[*]Equal contribution [1]CISPA Helmholtz Center for Information Security, Saarbrücken, Germany. Correspondence to: Sascha Xu <sascha.xu@cispa.de>, Nils P. Walter <nils.walter@cispa.de>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

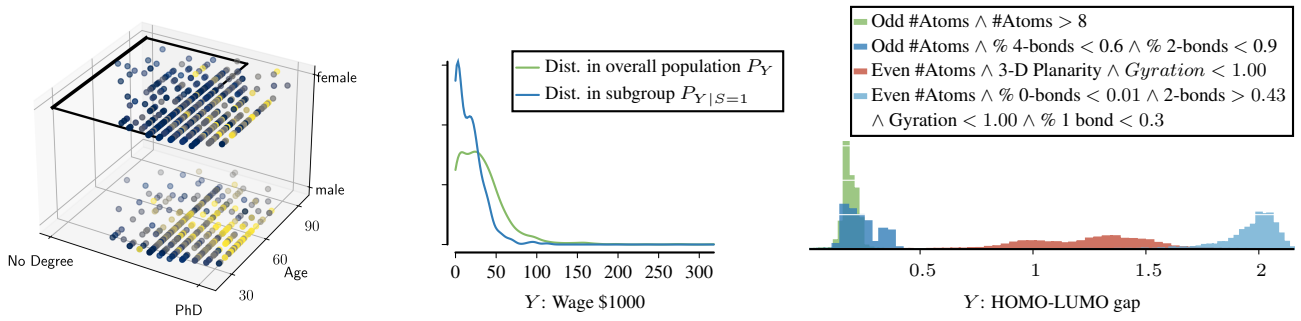
The common denominator in such applications is to present the relevant subgroups to a domain expert. In other words, not only do we require to find subsets with exceptional behavior, but also, that these can clearly be interpreted by the respective audience. That is, we have a joint optimization task of learning simple descriptions of sub-populations for which the property of interest is (locally) exceptionally distributed compared to the rest of the dataset. Typically, such a description is a conjunction of predicates, each based on the features of the dataset. For example on census data, where *wage* is the target property, a subgroup could be "*Women without higher education*" (Fig. 1a) have an exceptionally low salary compared to the overall population (Fig. 1b).

Since the introduction of subgroup discovery by Klösgen, many approaches have been proposed (Atzmueller, 2015). However, these have not kept up with the recent advances in machine learning and suffer from three main limitations. First, due to combinatorial optimization, these methods are limited to small datasets. Second, most methods assume that the target follows a simple distribution, e.g. normal or binomial distribution. Although there are proposals to learn a proxy of the target distribution, their results are less interpretable. Third, existing methods require a pre-quantization of the continuous features, which is independent of the optimization procedure. As we show in our experiments, this greatly influences the quality of the results.

To overcome these limitations, we propose SYFLOW. Our key contributions are as follows:

(i) We formulate subgroup discovery as a continuous optimization problem based on KL-divergence. This enables first-order optimization, which significantly improves runtime and performance.
(ii) We leverage Normalizing Flows (Rezende & Mohamed, 2015a) to accurately learn the target distribution from data, enabling us to deal with intricate real-world distributions.
(iii) We propose a neuro-symbolic rule layer to learn interpretable subgroup descriptions and the corresponding discretization in an end-to-end fashion.

We extensively evaluate SYFLOW on synthetic and real-world data. We show that SYFLOW accurately and reliably learns and characterizes exceptional subgroups, even for complex target distributions. We demonstrate that in con-



(a) *Wages* Subgroup S: "*Women without higher education*" earn significantly less.

(b) Exceptional distribution in subgroup (blue) vs. overall (green).

(c) SYFLOW learns subgroups of gold molecules with exceptional electron configurations.

Figure 1. Subgroups. SYFLOW learns subgroups, named subpopulations of which the distribution of the target variable is exceptional. In (**a**) SYFLOW precisely describes the subgroup of "*Women without higher education*", whose distribution of the target quantity *wage* is significantly lower (**b**). In general, SYFLOW is applicable on any data with non-trivial target distributions, e.g. material science (**c**).

trast to the state-of-the-art baselines, SYFLOW identifies diverse sets of subgroups. To showcase SYFLOW's strength on real-world tasks, we perform a case study in the domain of materials science, where we search for characteristics of gold clusters with exceptional conductivity (Fig. 1c) and reactivity. In both cases, SYFLOW identifies physically meaningful subgroups and their respective descriptions.

2. Preliminaries

We consider a dataset of n pairs $(\mathbf{x}, y)$, where $y \in \mathcal{Y}$ represents a property of interest, the **target property**, and $\mathbf{x} \in \mathbb{R}^m$ is a **feature vector**. From a statistical perspective, we assume $(\mathbf{x}, y)$ is a realization of a pair of random variables $(\mathbf{X}, Y) \sim \mathbb{P}(\mathbf{X}, Y)$. We denote random variables by capitals, write p for their density, and P for their law.

We are interested in learning **subgroups** for which the conditional distribution of the target attribute $P_{Y|S=1}$ is exceptional compared to P_Y. A subgroup membership function, or **rule**, $\sigma(\mathbf{x}) \in \{0, 1\}$ determines whether a sample x belongs to the subgroup (1) or not (0). Formally, it is a conjunction $\sigma : \mathbf{x} \mapsto \wedge_{i=1}^{m} \pi(x_i)$ of Boolean-valued predicates $\pi : \mathbb{R} \to \{0, 1\}$, where each predicate defines an interval over which its output is 1, e.g. "$18 < \text{age} < 65$".

To permit continuous optimization, we consider **soft predicates** $\hat{\pi} \in [0, 1]$. These are smooth functions that model the probability of a sample x to be inside an interval $\alpha < x < \beta$. We can control the steepness of the transition via a temperature parameter t. We write $s(x) \in [0, 1]$ to denote a **soft rule** based on soft predicates, and $S \in \{0, 1\}$ for the indicator (random) variable that s defines, i.e. $\mathbb{P}(S = 1 \mid \mathbf{X} = x) = s(x)$. Note that by reducing t to 0 we again obtain binary predicates and rules that are easy to interpret.

3. Method

In this section, we introduce SYFLOW for learning exceptional subgroups by end-to-end maximization of KL-divergence. We first give an overview and then the details.

3.1. Overview

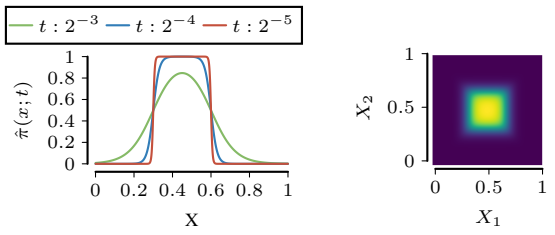
A subgroup is characterized by a membership function σ, which is commonly constrained to be a directly interpretable rule σ, i.e. a logical conjunction over predicates π. Finding the rule that identifies the most exceptional conditional distribution of the target is an NP-hard combinatorial problem (Boley & Grosskreutz, 2009).

We propose to take a different, end-to-end optimizable approach to learning subgroups. To this end, we propose a continuous relaxation of the binary-valued rule function that is designed to be differentiable, avoiding the need for pre-discretization, yet giving directly interpretable results. We propose to optimize the exceptionality of a subgroup in terms of KL-divergence between the conditional distribution $P_{Y|S=1}$ and the marginal distribution P_Y of the target, where we model these distributions non-parametrically using Normalizing Flows – with the added benefit that we have a single solution for univariate or multivariate targets.

As our entire framework is differentiable, we can optimize all components with gradient descent, which as we will see is often both faster and more performant than combinatorial approaches. Finally, our framework naturally enables iteratively learning multiple non-redundant subgroups by regularizing with the similarity of already learned subgroups.

3.2. Differentiable Rule Induction

Subgroup membership $\sigma : \mathbf{x} \mapsto \wedge_{i=1}^{m} \pi(x_i; \alpha_i, \beta_i)$ is defined by a logical conjunction over binary predicates π,



(a) Soft predicate $\hat{\pi}(x;t)$ for different temperatures t.

(b) Subgroup membership probability for soft rule $s(x)$.

Figure 2. Example soft predicate **(a)** and soft rule **(b)**.

where we use one predicate per feature. Predicates for features that are not relevant for a subgroup are always true.

Our key idea is to redefine the subgroup membership function in probabilistic terms, such that we obtain a function that acts akin to logical conjunctions while at the same time it is differentiable and therewith can be learned using a gradient-descent-based method. In particular, we propose to associate each feature X_i with a **soft predicate** π_i

$$\hat{\pi}(x_i;\alpha_i,\beta_i,t) = \frac{e^{\frac{1}{t}(2x_i-\alpha_i)}}{e^{\frac{1}{t}x_i} + e^{\frac{1}{t}(2x_i-\alpha_i)} + e^{\frac{1}{t}(3x_i-\alpha_i-\beta_i)}} \quad (1)$$

where we adapt the idea of approximate and differentiable splits from deep decision trees (Yang et al., 2018), we use a lower and upper bound $\alpha, \beta \in \mathbb{R}$, and introduce a temperature parameter $t > 0$ that controls the steepness of the function at these bounds. The lower the temperature, the less soft the predicate, and in the limit of $t \to 0$ a soft predicate converges to a strict predicate. In Fig. 2a we show an example soft predicate for different temperatures. For smaller t, the soft predicate closely approximates the strict predicate, except for a negligible region around the bounds.

Theorem 1 *Given its lower and upper bounds $\alpha_i, \beta_i \in \mathbb{R}$, the soft predicate of Eq. (1) applied on $x \in R$ converges to the crisp predicate that decides whether $x \in (\alpha, \beta)$,*

$$\lim_{t\to 0} \hat{\pi}(x_i;\alpha_i,\beta_i,t) = \begin{cases} 1 & \text{if } \alpha_i < x_i < \beta_i \\ 0.5 & \text{if } x_i = \alpha_i \vee x_i = \beta_i \\ 0 & \text{otherwise} \end{cases} .$$

We provide the proof for the general case in Appendix A.

The soft predicate $\hat{\pi}$ provides a differentiable, adaptable binning function. Next, we propose to combine the predicates $\hat{\pi}$ for each feature x_i into a soft rule s that acts akin to logical conjunctions but remains differentiable. It is possible to model a logical conjunction using multiplication, but this leads to vanishing gradients (Hochreiter, 1998), which is problematic especially for non-trivial amounts of soft predicates (features). The harmonic mean

$$\mathcal{M}(x) = \frac{p}{\sum_{i=1}^{m} \hat{\pi}(x_i;\alpha_i,\beta_i,t)^{-1}} ,$$

behaves as desired for strictly binary predicates, i.e. $\exists\hat{\pi}(x_i;\alpha_i,\beta_i,t) = 0 \Rightarrow \mathcal{M}(x) = 0$ and $\forall\hat{\pi}(x_i;\alpha_i,\beta_i,t) = 1 \Rightarrow \mathcal{M}(x) = 1$, but tends to break down when given many soft predicates, e.g. for a very high dimensional feature space. To avoid this, we propose to instead use the *weighted harmonic mean* to model logical conjunctions, and construct the soft rule function s as

$$s(x;\alpha,\beta,a,t) = \frac{\sum_{i=1}^{m} a_i}{\sum_{i=1}^{m} a_i \hat{\pi}(x_i;\alpha_i,\beta_i,t)^{-1}} . \quad (2)$$

The weights $a \in \mathbb{R}^m$, which are constrained to be positive through a ReLU function, allow the conjunction layer to disable unnecessary predicates. Wherever $a_i > 0$, the weights do not affect behavior for strictly binary predicates.

In Fig. 2b, we show an example subgroup membership function for a soft rule s . The subgroup here is characterized by a conjunction of two predicates on X_1 and X_2, i.e. a box, with a gradual, smooth transition from subgroup inclusion to exclusion at the boundaries.

In general, our formulation of a soft rule is completely flexible in regards to the thresholds of the binned features, and for $t \to 0$ is asymptotically equivalent to a strict rule.

3.3. Differentiable Density Estimation

Besides a differential rule function, we require accurate estimations of the conditional resp. marginal distributions of the target. We deviate from existing work by taking a differentiable non-parametric approach in the form of *Normalizing Flows*. These are an increasingly popular class of density estimators (Rezende & Mohamed, 2015b; Dinh et al., 2017; Papamakarios et al., 2021). The fundamental idea behind a normalizing flow is to start with a distribution with a known density function, e.g. a Gaussian distribution with $p_{\mathcal{N}}$, and fit an invertible function f to transform it onto the target density.

Our method, SYFLOW, allows to seamlessly use any normalizing flow architecture. In this work, our architecture of choice are Neural Spline Flows (Durkan et al., 2019), which use expressive yet invertible piece-wise, polynomial spline functions. In general, the idea is to train the function f so that $p_{\mathcal{Y}} \approx f(p_{\mathcal{N}})$. Given a sample y, we can compute the likelihood of that point under the current function f as $p_{f(\mathcal{N})}(y) = p_{\mathcal{N}}(f^{-1}(y))|\det\left(\frac{\delta f^{-1}(y)}{\delta y}\right)|$. Thus, given a sample of $\mathbb{P}(Y)$, we can maximize the likelihood of $p_{f(\mathcal{N})}(y)$ and hence fit $p_{f(\mathcal{N})} \approx p_{\mathcal{Y}}$.

3.4. Differentiable Exceptionality Measure

We now have a differentiable rule function s and versatile density estimate p_Y resp. $p_{Y|S=1}$ of the target distribution. Next, we propose a differentiable measure of exceptionality between the conditional target distribution and the marginal

target distribution. We adopt the Kullback-Leibler (KL) divergence and show how we can reformulate it towards this goal. We start with the standard definition,

$$D_{\mathrm{KL}}\left(P_{Y|S=1} \| P_Y\right) = \int_{y \in \mathcal{Y}} p_{Y|S=1}(y) \log\left(\frac{p_{Y|S=1}(y)}{p_Y(y)}\right) dy \,. \tag{3}$$

Here, the soft rule s does not explicitly appear, although we need it to take gradients wrt. the parameters. Towards this, we rewrite the first occurrence of $p_{Y|S=1}$ in Eq. (3) as

$$\begin{aligned} p_{Y|S=1}(y) &= \int_{\mathbf{x} \in \mathbb{R}^m} p_{Y|S=1,\mathbf{x}}(y,\mathbf{x}) p_{\mathbf{x}|S=1}(\mathbf{x}) d\mathbf{x} \\ &= \int_{\mathbf{x} \in \mathbb{R}^m} p_{Y|S=1,\mathbf{x}}(y,\mathbf{x}) \frac{p_{S=1|\mathbf{x}}(\mathbf{x}) p_{\mathbf{x}}(\mathbf{x})}{\mathbb{P}\left(S=1\right)} d\mathbf{x} \,, \end{aligned} \tag{4}$$

by using the rules of marginal probability resp. Bayes' rule. We will first approximate Eq. (4), and then show how to estimate the KL divergence of Eq. (3) for its optimization.

To this end, we first note that the subgroup indicator S takes two discrete values, indicating whether $\mathbf{x}$ belongs to the subgroup. The rule function $s(\mathbf{x})$ is deterministic in the limit of $t \to 0$ as per Theorem 1. We use this to partition the domain of integration $\mathbb{R}^m$ into $\mathbb{R}^m_{\in} := \{\mathbf{x} \in \mathbb{R}^m | s_{t\to 0}(\mathbf{x}) = 1\}$ and $\mathbb{R}^m_{\notin} := \{\mathbf{x} \in \mathbb{R}^m | s_{t\to 0}(\mathbf{x}) = 0\}$. Under the following four assumptions, we can bound the approximation of density $p_{Y|S=1}$ in Eq. (4). First, we assume that both $p_{\mathbf{x}}$ and $p_{Y|S=1,\mathbf{x}}$ are upper bounded by the finite constants $C_{\mathbf{X}} > 0$ and $C_Y > 0$, respectively. Secondly, we assume that in a subset $\mathbb{R}^m_{\notin\subset} \subset \mathbb{R}^m_{\notin}$, where s is neither zero or one, i.e. the yellow region in Figure 2b, is negligible. Lastly, we assume that $\mathbb{R}^m_{\notin\subset}$ covers almost all of the non-membership domain $\mathbb{R}^m_{\notin}$ and the probability mass in this area is also negligible. Formally, we assume

$$p_{\mathbf{x}}(\mathbf{x}) \leq C_{\mathbf{X}}, \tag{5}$$

$$p_{Y|S=1,\mathbf{x}} \leq C_Y, \tag{6}$$

$$\int_{\mathbf{x} \in \mathbb{R}^m_{\notin\subset}} p_{S=1|\mathbf{x}}(\mathbf{x}) d\mathbf{x} \leq \epsilon_1 \,, \tag{7}$$

$$\int_{\mathbf{x} \in \mathbb{R}^m_{\notin} \setminus \mathbb{R}^m_{\notin\subset}} p_{\mathbf{x}}(\mathbf{x}) d\mathbf{x} < \epsilon_2 \,. \tag{8}$$

Theorem 2 *When Eqs. (5), (6), (7), and (8) hold, it is*

$$p_{Y|S=1}(y) - \int_{\mathbf{x} \in \mathbb{R}^m_{\in}} p_{Y|S=1,\mathbf{x}}(y,\mathbf{x}) d\mathbf{x} \leq \frac{C_Y(\epsilon_2 + C_{\mathbf{X}}\epsilon_1)}{\mathbb{P}\left(S=1\right)} \,.$$

Further, during learning, this bound becomes tighter until it asymptotically vanishes, assuming a decreasing annealing schedule for the temperature parameter.

We postpone the proof to Appendix B.

For an element of the subgroup $\mathbf{x} \in \mathbb{R}^m_{\in}$ $p_{S=1|x}$ is either zero or one, up to a negligible region (Ass. 7). We further approximate the target property conditional for $\mathbf{x} \in \mathbb{R}^m_{\in}$

$$\begin{aligned} p_{Y|\mathbf{x}}(y,\mathbf{x}) &= p_{Y|S=1,\mathbf{x}}(y,\mathbf{x}) p_{S=1|\mathbf{x}}(\mathbf{x}) \\ &\quad + p_{Y|S=0,\mathbf{x}}(y,\mathbf{x}) p_{S=0|\mathbf{x}}(\mathbf{x}) \\ &\approx p_{Y|S=1,\mathbf{x}}(y,\mathbf{x}), \end{aligned}$$

The approximation follows from a case distinction on $p_{S=1|x}$. This allows us to approximate the subgroup-conditional target distribution from Eq. (4) as

$$\begin{aligned} p_{Y|S=1}(y) &= \int_{\mathbf{x} \in \mathbb{R}^m_{\in}} p_{Y|S=1,\mathbf{x}}(y,\mathbf{x}) \frac{p_{S=1|\mathbf{x}}(\mathbf{x}) p_{\mathbf{x}}(\mathbf{x})}{\mathbb{P}\left(S=1\right)} dx \\ &\approx \int_{\mathbf{x} \in \mathbb{R}^m_{\in}} p_{Y,\mathbf{x}}(y,\mathbf{x}) \frac{p_{S=1|\mathbf{x}}(\mathbf{x})}{\mathbb{P}\left(S=1\right)} dx \,. \end{aligned} \tag{9}$$

Finally, we replace Eq. (9) into Eq. (3) to obtain our final approximation

$$D_{\mathrm{KL}}\left(P_{Y|S=1} \| P_{Y|S=1}\right) \approx \int_{y \in \mathcal{Y}} \int_{\mathbf{x} \in \mathbb{R}^m_{\in}} p_{Y,\mathbf{x}}(y,\mathbf{x}) \frac{p_{S=1|\mathbf{x}}(\mathbf{x})}{\mathbb{P}\left(S=1\right)} dx \log\left(\frac{p_{Y|S=1}(y)}{p_Y(y)}\right) dy \,.$$

From this point onward, we can use the standard Monte Carlo estimation of this integral. This gives

$$D_{\mathrm{KL}}\left(P_{Y|S=1} \| P_Y\right) \approx \frac{1}{n_s} \sum_{k=1}^{n} s(\mathbf{x}^{(k)}) \log\left(\frac{p_{Y|S=1}(y^{(k)})}{p_Y(y^{(k)})}\right) \,,$$

where p_Y and $p_{Y|S=1}$ stand for the models trained from the normalizing flows, s is our subgroup membership model (see Sec. 3.2) and n_s is estimated as $\frac{1}{n}\sum_{i=1}^{n} s(\mathbf{x}^{(i)})$. Our approximation is directly computable given the density estimates p_Y and $p_{Y|S=1}$ from the normalizing flows. Crucially this allows us to update the subgroups rule s as to maximize its exceptionality/KL-Divergence. Finally, we can now deal with some fine tuning of the objective to discover both representative and diverse subgroups.

3.5. Rule Generality and Diversity

The KL-divergence measures dissimilarity between two distributions. Naively maximizing it, however, has a drawback as we could easily craft a small subgroup consisting of the single most deviating sample on its own, defined by a rule with a very narrow scope and relatively low value. Thus, we employ a common technique in subgroup discovery (Boley et al., 2017) in order to steer the results towards larger subgroups: we multiply the statistic of dissimilarity with the size of the subgroup n_s^{γ}. The power γ tunes the trade-off in the importance of subgroup exceptionality and size.

As we will show in our experiments, traditional subgroup discovery approaches often find sets of nearly identical subgroups. In the typical top-k scheme, the best-scored subgroups are often slight variations of the same rule. To encourage SYFLOW to learn subgroups with diverse distributions, we introduce a **regularizer**, based on the KL-divergence of the current subgroup S to the previously found subgroups S_k to our objective. Hence, summarizing all the above, we obtain as our final objective our variant of the **size-corrected KL** (van Leeuwen & Knobbe, 2011):

$$D_{\mathrm{WKL}}\left(P_{Y|S=1} \| P_Y\right) = \\ n_s^\gamma \hat{D}_{\mathrm{KL}}\left(P_{Y|S=1} \| P_Y\right) + \lambda \sum_{j=1}^{k} \hat{D}_{\mathrm{KL}}\left(P_{Y|S=1} \middle\| P_{Y|S_j=1}\right) . \tag{10}$$

The parameter λ controls the strength of the regularizer.

3.6. Full Model

In the previous sections, we detailed our rule learning architecture with differentiable thresholding and aggregation (Sec. 3.2). We described how to use Neural Spline Flows to obtain non-parametric density estimates (Sec. 3.3), and finally derived Objective (10), a size aware Kullback-Leibler Divergence that allows us to optimize our rule function $s(\mathbf{x})$ with gradient descent. Together these make up the components of our SYFLOW architecture for subgroup discovery with normalizing flows. Given a dataset $\{(\mathbf{x}^{(k)}, y^{(k)})\}_{k=1}^{N} \sim \mathbb{P}(\mathbf{X}, Y)$, SYFLOW undergoes the following three steps for each sample $(\mathbf{x}^{(k)}, y^{(k)})$:

1. *Feature Thresholding*: All continuous features $\mathbf{x}_i^{(k)}$ are thresholded with learned parameters α_i and β_i using the soft-binning from Eq. (2). Thereby we obtain a predicate vector $\hat{\pi}(\mathbf{x}^{(k)}; \alpha, \beta, t) \in [0,1]^p$.

2. *Subgroup Rule*: We employ weights a_i to combine the individual predicates $\hat{\pi}(\mathbf{x}_i^{(k)})$ into a conjunction $s(\mathbf{x}; \alpha, \beta, a, t)$. This rule represents the probability of subgroup membership $\mathbb{P}(S = 1 | X = \mathbf{x}^{(k)})$.

3. *Distribution Exceptionality*: We estimate the likelihood of $p_Y(y^{(k)})$ and $p_{Y|S=1}(y^{(k)})$ with two separately fitted normalizing flow models. Then, according to Objective (10), we can estimate the KL-Divergence between the current subgroup and the general distribution.

By repeating the aforementioned steps over all samples $(\mathbf{x}^{(k)}, y^{(k)})$ and summing up the results, Objective (10) gives us a differentiable estimate of the KL-Divergence in regards to the subgroup rule $s(\mathbf{x})$. We optimize $s(\mathbf{x})$ using standard gradient descent techniques with the Adam optimizer (Kingma & Ba, 2015). After the subgroup rule has been updated, we again update the normalizing flow of the subgroups density as described in Sec. 3.3, and repeat this process for a user-specified amount of epochs. During the training, we gradually decrease the temperature t by a pre-determined schedule to obtain increasingly crisp subgroup assignments. Finally, at the last epoch, the discovered subgroup is then the output of the subgroup rule $s(\mathbf{x})$. We provide a diagram overviewing and the pseudo-code for SYFLOW in the Appendix C.

4. Related Work

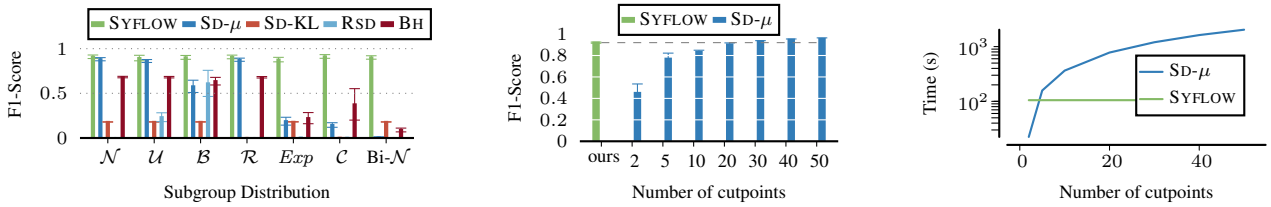
Subgroup Discovery. Traditional approaches for subgroup discovery (Klösgen, 1996) can be split based on type of search and on exceptionality measures. Subgroup discovery is NP-hard (Boley & Grosskreutz, 2009) and hence most proposals resort to greedy heuristics (Duivesteijn et al., 2016; Atzmueller, 2015) without guarantees. Branch-and-bound based algorithms (Boley et al., 2017; Kalofolias et al., 2019) permit results with guarantees for some exceptionality measures, but generally do not scale beyond tens of features (Atzmueller & Puppe, 2006).

Most proposals for subgroup discovery assess exceptionality by comparing the conditional and marginal distributions of a single univariate target (Song et al., 2016; Helal, 2016). Basic measures of exceptionality include mean-shift (Grosskreutz & Rüping, 2009) for continuous, and weighted relative accuracy (Song et al., 2016) for discrete targets, but there exists a wide range of proposals for many data types (Kalofolias & Vreeken, 2022). Most, however, make strong assumptions about the distribution of the target such as normal (Friedman & Fisher, 1999; Lavrač et al., 2004), binomial or χ^2 (Grosskreutz & Rüping, 2009).

Duivesteijn et al. (2016) generalize subgroup discovery to multivariate targets by measuring exceptionality via the difference between models just trained on the subgroup versus on all of the data. Due to computational costs, only simple models can be used, leading to a compromise in performance. Proença et al. (2022) instead measure exceptionality using a proxy of KL-divergence based on the Minimum Description Length (MDL). In contrast, SYFLOW optimizes KL-divergence directly, can employ any type of normalizing flow, and is equally suited for uni-/multi-variate targets.

Neural Set Functions (NSFs) (Zaheer et al., 2017; Ou et al., 2022) are similar in spirit to subgroup discovery, as they also seek to find exceptional points. However, they significantly differ in two aspects. First, NSFs are learned using (partial) supervision, while subgroup discovery is unsupervised. Second, subgroup discovery finds interpretable descriptions, while NSFs come in a black-box.

Subgroup discovery is also related to *slice discovery* (Chung et al., 2019; d'Eon et al., 2022; Eyuboglu et al., 2022) which



(a) F1-scores of recovering a planted subgroup for different target distributions, higher is better.

(b) F1-scores for SYFLOW and SD-μ, higher is better.

(c) Runtime of SYFLOW and SD-μ, lower is better.

Figure 3. Subgroup Predictive Accuracy. **(a)** Method comparison in terms of F1-score recovering subgroups in synthetic data. **(b)** Across different distributions: SYFLOW outperforms the competition on distributions with higher order moments. **(c)** With increasing number of cutpoints, SD-μ matches SYFLOW accuracy around 40 bins, but needs 10× more time.

aims to find subsets of the data where a deep learning model has high error rates. Slice discovery methods return membership functions defined on the latent space of the network. The focus lies on discovering concept-aligned slices rather than explicit rules on latent features. This makes slice discovery methods not applicable for generic subgroup discovery tasks. In contrast, methods such as SYFLOW can be useful for slice discovery with appropriate modifications.

Differentiable Rule Induction Classical rule induction methods aim to find rules of the form " **if** $X_1 = 1 \wedge X_5 = 1$ **then** $Y = 0$" through expensive combinatorial optimization. Recently, highly scalable differentiable rule induction methods were proposed based on differentiable analogues of logical connectives, e.g. conjunctions and disjunctions (Fischer & Vreeken, 2021; Wang et al., 2020) that permit extracting crisp logical rules after training. Most work in this direction focuses on learning a global classifier (Yang et al., 2018; Qiao et al., 2021; Wang et al., 2021; Dierckx et al., 2023) as opposed to our goal of learning concise rules that identify exceptional subgroups. In this sense most related is Walter et al. (2024), who propose a neural architecture to find conjunctions of binary features that are over resp. under-expressed for a particular label. In contrast, SYFLOW considers continuous features, and is not constrained to a type or number of target variables.

5. Experiments

We evaluate SYFLOW against four state-of-the-art methods on synthetic and real-world data. We compare against Bump Hunting (BH, Friedman & Fisher, 1999), subgroup discovery using mean-shift (SD-μ, Lemmerich & Becker, 2018), subgroup discovery using KL-divergence (SD-KL), and Robust Subgroup Discovery (RSD, Proença et al., 2022). We give the hyperparameters in Appendix D and provide the data generators as well as the code online.[1]

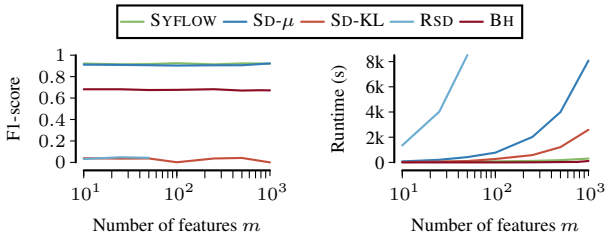
[1] https://eda.rg.cispa.io/prj/syflow/

5.1. Synthetic Data

To evaluate on datasets with known ground truth we consider synthetic data. We begin by generating m feature variables $\mathbf{X}_i$ and the target variable Y from a uniform distribution $\mathcal{U}(0,1)$ to create a dataset of $n = 20\,000$ samples. Within this dataset, we plant a rule $\sigma(x) = \wedge_i^c \pi(x_i; \alpha_i, \beta_i)$ of c conditions. The hypercube described by the rule is set to have a volume of 0.1, i.e. 10% of the population. For the samples within the planted subgroup, we re-sample the target variable Y using a separate distribution $\mathbb{P}(Y \mid S = 1)$. We run each experiment five times and report the average.

Target Distribution First, we assess for all methods their accuracy in recovering the planted subgroup for different distributions of the target property Y. To this end, we vary $\mathbb{P}(Y \mid S = 1)$ to be respectively a normal distribution $\mathcal{N}(1.5, 0.5)$, uniform distribution $\mathcal{U}(0.5, 1.5)$, exponential distribution $\text{Exp}(0.5)$, Rayleigh distribution $\mathcal{R}(2)$, Cauchy distribution $\mathcal{C}(0,1)$, beta distribution $\mathcal{B}(0.2, 0.2)$, and a balanced mixture distribution of two gaussians (Bi-$\mathcal{N}$) $\mathcal{N}(-1.5, 0.5)$ and $\mathcal{N}(1.5, 0.5)$. The distribution are shown in Figure 8 in Appendix G. The number of features $m = 10$ and complexity $c = 4$ remains fixed.

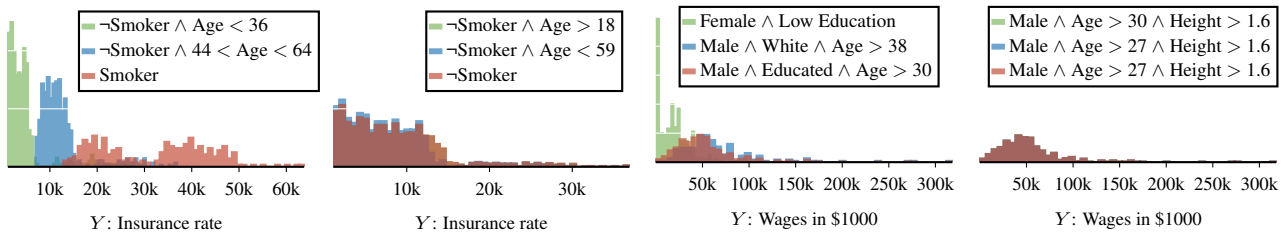
For each method, we compute the F1-score between the ground truth and discovered subgroup labels (i.e. $S = 1$). We present the results in Figure 3a; We see that for distri-



(a) F1 under increasingly more features m, higher is better.

(b) Runtime under increasingly more features m, lower is better.

Figure 4. Scalability of SYFLOW and baselines.



(a) *Insurance*: **SYFLOW** rules (b) *Insurance*: **SD-KL** rules (c) *Wages*: **SYFLOW** rules (d) *Wages*: **SD-μ** rules

Figure 5. Subgroups learned on the *Insurance* and *Wages* datasets. Only SYFLOW learns diverse and exceptional subgroups.

butions that are well characterized by their first moment, i.e. the uniform and normal distribution, SYFLOW, SD-μ and BH are all able to recover the planted subgroup. On the exponential, Rayleigh and bi-modal distributions, only SYFLOW is able to recover the majority of the planted subgroup. In general, SYFLOW reliably recovers subgroups independent of the underlying target distribution.

Thresholding Next, we study the efficacy of the differentiable feature thresholding. We generate data as before, considering only Normal distributions for the target variable Y, and setting $m = 50$. We compare the accuracy of SYFLOW against SD-μ, whilst gradually increasing the amount of bins per feature in the pre-processing for SD-μ.

As we can see in Figure 3b, as the number of cutpoints increases, the F1-score of SD-μ improves. Under 20 cutpoints SD-μ performs much worse than SYFLOW, while for more cutpoints it slightly outperforms it in terms of accuracy. At the same time, as the number of cutpoints increases, SD-μ runtime increases rapidly (Figure 3c), requiring an order of magnitude more runtime to perform on par with SYFLOW.

Scalability In the final experiment on synthetic data, we assess performance and runtime when varying the number of features. We allow each method up to 24 hours. We present our results in Figure 4. We observe that all methods perform remarkably stable in terms of F1-score, as well as that SYFLOW is significantly faster than all of its competitors. As a continuous optimization based method, SYFLOW avoids the typical combinatorial explosion in runtime, and additionally takes advantage of GPU acceleration. In comparison, SD-μ, SYFLOWS closest competitor, takes 50 times longer for 1 000 features, whereas RSD does not finish within 24 hours for more than 100 features.

Hyperparameter Sensitivity Lastly, we study the sensitivity of SYFLOW to its hyperparameters. We report the average F1-score over all target distributions, varying each hyperparameter individually whilst keeping the others fixed as in the previous experiments.

For γ, which controls the trade-off between size and exceptionality, SYFLOW works well in a range of $\gamma \in [0.4, 1]$, with an average F1 score of 0.85 upwards. For the regularizer λ, we observe a 0.05 improvement when it is active, whilst varying its strength between $[0.5, 3]$ does not have a noticeable impact. Finally, for the predicate temperature t that controls the softness of the binning, the ideal range by empirical observation lies in between $[0.01, 0.2]$. In both synthetic and real-world experiments we used a temperature of $t = 0.2$ which provides good performance across the board. We provide a detailed analysis in Appendix D.

5.2. Real World Data

We now turn to real-world data, where we evaluate on regression datasets from the UCI-Machine Learning Repository.[2] They provide a variety of target distributions and feature spaces to search and allow us to qualitatively inspect the learned rules. On each dataset, we let each method return 5 subgroups. For SYFLOW we report the average over 100 runs and report the standard deviation in Tab. 3. As the ground truth is unknown, we assess the subgroup quality in terms of KL-divergence (D_{KL}), the distribution overlap or Bhattacharyya Coefficient (BC), and by absolute difference in mean (AMD). As D_{KL} and AMD are strongly influenced by the size n_s of the subgroup, we correct for this. For a formal definition we refer to Appendix E. In Table 1 we report, per metric, the scores of the best subgroup each method found.

Across the board, SYFLOW reliably finds exceptional subgroups, and is either the best or close to the best method when it comes to the distribution based exceptionality measures (D_{KL} and BC). SD-μ optimizes for mean difference, and hence it is not surprising that it outperforms SYFLOW on this metric, but that SYFLOW nevertheless comes so close in terms of scores is a very positive result indeed.

For the *Insurance* dataset it is inconclusive which method learns the most exceptional subgroups; SYFLOW returns the best subgroup in terms of D_{KL}, SD-KL finds the subgroup with the largest mean difference. For further analysis, we

[2] https://archive.ics.uci.edu.

Table 1. Quantitative results of exceptionality of the subgroups discovered by resp. SYFLOW (ours), SD-KL, SD-μ, RSD, and BH, as measured KL-Divergence (D_{KL}, higher is better), Bhattacharyya coefficient (BC, lower is better), and absolute mean distance (AMD, higher is better).

	D_{KL}					BC					AMD				
	ours	SD-KL	SD-μ	RSD	BH	*ours*	SD-KL	SD-μ	RSD	BH	*ours*	SD-KL	SD-μ	RSD	BH
Abalone	**0.15**	0.02	0.12	0	0.05	**0.69**	0.99	0.93	1	0.87	0.73	0.25	**0.84**	0	0.16
Airquality	0.22	0.22	**0.24**	0	0.0	**0.65**	0.86	0.79	1	1.0	0.37	**0.53**	0.49	0	0.0
Automobile	0.25	0.24	0.23	**0.26**	0.21	0.63	0.85	0.79	0.64	**0.6**	1838	**2807**	2683	2218	2476
Bike	**0.17**	0.1	0.15	0.17	0.13	**0.64**	0.95	0.9	0.67	0.73	584	570	**630**	432	622
California	**0.13**	0.06	0.11	0	0.0	**0.74**	0.97	0.93	1	1.0	0.25	0.3	**0.32**	0	0.0
Insurance	**0.26**	0.13	0.26	0	0.19	**0.41**	0.93	0.52	1	0.84	3846	**3973**	3845	0	1518
Mpg	**0.27**	0.26	0.24	0.21	0.24	0.57	0.76	0.8	**0.47**	0.61	**2.99**	2.85	2.96	1.66	2.79
Student	0.08	0.03	0.08	**0.09**	0.04	0.85	0.99	0.94	**0.71**	0.97	0.46	0.52	**0**	0.47	0.45
Wages	**0.1**	0.02	0.1	0	0.03	**0.88**	0.99	0.9	1	0.99	**6043**	2994	5916	0	5149
Wine	**0.06**	0.0	0.06	0	0.01	**0.9**	1.0	0.97	1	0.97	0.17	0.04	**0.19**	0	0.04
Avg. rank	**1.3**	3.6	2.0	3.5	3.6	**1.3**	4.0	2.9	3.4	2.9	2.6	2.4	**1.5**	4.5	3.6

plot the best three subgroups found by SYFLOW and SD-KL in Figure 5a and 5b. The specific rules that SYFLOW finds are succinct and informative (¬Smoker $\wedge$ Age $<$ 36), and represent a diverse set of subgroups (low, medium and high premiums). In contrast, SD-KL finds highly redundant rules, e.g. "¬Smoker" and "¬Smoker $\wedge$ Age $>$ 18", that all describe the same subpopulation.

Generally, on all datasets where the target variable is exponentially distributed (*Wages*, *Insurance*, *California*), SYFLOW performs well in any measured metric, which matches the results seen on the exponential synthetic data (Sec. 5.1). Finally, in Fig. 5c and 5d, we plot the learned subgroups on the *Wages* dataset mentioned in the introduction. Again, SYFLOW finds diverse subgroups of disadvantaged demographics (Female $\wedge$ Low Education) and advantaged groups (Male $\wedge$ White $\wedge$ Age $>$ 38), whilst its competitor find either much less exceptional subgroups (RSD,BH) or variants of the same subgroup (SD-μ,SD-KL).

Overall, SYFLOW shows versatility in finding exceptional subgroups across a variety of datasets and metrics. The discovered rules are succinct and diverse, can scale to large datasets, and are robust to the underlying target distribution. In the following, we will use SYFLOW on a specific application in materials science.

5.3. Case Study: Materials Science

Next, we consider a case study on materials science data (Goldsmith et al., 2017), an application where learning diverse exceptional subgroups has a clear scientific benefit. In particular, we consider a dataset of properties of gold-nano-clusters. These are key components in photo-voltaics, as well as in medical applications (Giljohann et al., 2020). The key goal is to better understand which molecular configurations lead to materials with better properties in terms of absorbing photons or with which cells they interact.

We first focus on characterizing the HOMO-LUMO gap, the difference between the highest occupied and lowest unoccupied molecular orbit, which corresponds to the efficiency in energy absorption of the material for photons of specific frequencies. Gold nano-clusters are also increasingly used in medical applications, in which the HOMO-LUMO gap is crucial to determine their reactivity with other molecules.

We show the subgroups that SYFLOW learns for the HOMO-LUMO gap in Figure 1c. The discovered subgroups relate to known ground-truth factors (Goldsmith et al., 2017), i.e. odd number of atoms leads to low gaps, but also ones with more complex descriptors such as clusters with an even amount of atoms that are mostly made up of double bond connections. SYFLOW provides *out of the box* interpretable, exceptional and diverse subgroups on this complex target quantity.

Next, we additionally consider the relative intramolecular Van der Waals energies as the target quantity. These quantities are important to determine the strength of molecular binding, e.g. allow a compound to selectively interact with the intended target. We show the results in Figure 6. It is

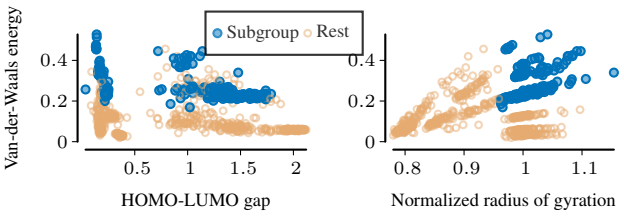


Figure 6. Gold Nano-Clusters. SYFLOW discovers a subgroup of molecules that have an outstanding *joint* distribution of Van-der-Waals energy *and* HOMO-LUMO gap.

easy to see that samples fitting subgroup rule s are indeed exceptional with regard to the overall distribution. Moreover, when we analyze the rule in detail, it makes physical sense. Molecules of between 8 to 14 atoms with a low (close to 1) gyration tend to have a significantly higher Van der Waals energy than those that are more strongly gyrated (i.e. are less 'flat') (Goldsmith et al., 2017).

6. Conclusion

We explored the problem of discovering a diverse set of exceptional subgroups, where the distribution of the target within each subgroup differs significantly from the overall population. Existing works suffer from combinatorial explosion, can not deal with intricate real world distributions, and heavily depend on the pre-quantization of the features. To overcome these limitations, we propose SYFLOW, where we formulate subgroup discovery as a continuous optimization problem. We use normalizing flows to accurately learn the distribution of the target property, enabling us to deal with arbitrary distributions. We propose a differentiable rule learner, which simultaneously learns the subgroup description and the corresponding discretization. We show on synthetic and real-world data, including a case study on gold nano-clusters, that SYFLOW reliably discovers diverting subgroups, especially when the target distribution is complex. In a case study on gold nano-clusters, we find subgroups that correspond to physically plausible processes.

Limitations

A current limitation of all subgroup discovery methods, including SYFLOW, is that the description language of conjunctions of Boolean predicates may be too simple to describe true subgroups for physical data. We are specifically interested in exploring how we can extend SYFLOW to perform symbolic regression. Furthermore, we are interested in adapting SYFLOW to deal with structured data, such as images. We provide first experiments in Appendix I, where we find that SYFLOW is capable of discovering subgroups corresponding to the digits 0 and 1 in the MNIST dataset. Whilst this at the moment requires extensive pre-processing, we will pursue future work to allow SYFLOW to directly work with structured data such as images.

Impact Statement

We propose a method, which can assist practitioners in making new scientific discoveries. For example, new insights into gold-cluster can have a potential impact on biomedical applications. However, when applied to sensitive census data, it is important to stress that SYFLOW is based on correlations and is not designed to make a definite causal statement. Thus, the results must be handled responsibly.

References

Atzmueller, M. Subgroup discovery. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 5(1): 35–49, 2015.

Atzmueller, M. and Puppe, F. SD-Map – A Fast Algorithm for Exhaustive Subgroup Discovery. In *Knowledge Discovery in Databases: PKDD 2006*, pp. 6–17. Springer Berlin Heidelberg, 2006.

Boley, M. and Grosskreutz, H. Non-redundant subgroup discovery using a closure system. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 179–194. Springer, Springer, 2009.

Boley, M., Goldsmith, B. R., Ghiringhelli, L. M., and Vreeken, J. Identifying Consistent Statements about Numerical Data with Dispersion-Corrected Subgroup Discovery. *Data Mining and Knowledge Discovery*, pp. 1391–1418, September 2017.

Boll, C. and Lagemann, A. The gender pay gap in eu countries—new evidence based on eu-ses 2014 data. *Intereconomics*, 54:101–105, 2019.

Chung, Y., Kraska, T., Polyzotis, N., Tae, K. H., and Whang, S. E. Slice finder: Automated data slicing for model validation. In *2019 IEEE 35th International Conference on Data Engineering (ICDE)*, pp. 1550–1553. IEEE, 2019.

d'Eon, G., d'Eon, J., Wright, J. R., and Leyton-Brown, K. The spotlight: A general method for discovering systematic errors in deep learning models. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, pp. 1962–1981, 2022.

Dierckx, L., Veroneze, R., and Nijssen, S. Rl-net: Interpretable rule learning with neural networks. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 95–107. Springer, 2023.

Dinh, L., Sohl-Dickstein, J., and Bengio, S. Density estimation using real nvp. In *International Conference on Learning Representations*, 2017.

Duivesteijn, W., Feelders, A. J., and Knobbe, A. Exceptional Model Mining: Supervised descriptive local pattern mining with complex target concepts. *Data Mining and Knowledge Discovery*, pp. 47–98, January 2016.

Durkan, C., Bekasov, A., Murray, I., and Papamakarios, G. Neural spline flows. *Advances in Neural Information Processing Systems*, 32, 2019.

Eyuboglu, S., Varma, M., Saab, K. K., Delbrouck, J.-B., Lee-Messer, C., Dunnmon, J., Zou, J., and Re, C. Domino: Discovering systematic errors with cross-modal embeddings. In *International Conference on Learning Representations*, 2022.

Fischer, J. and Vreeken, J. Differentiable pattern set mining. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pp. 383–392, 2021.

Friedman, J. H. and Fisher, N. I. Bump hunting in high-dimensional data. *Statistics and computing*, 9(2):123–143, 1999.

Giljohann, D. A., Seferos, D. S., Daniel, W. L., Massich, M. D., Patel, P. C., and Mirkin, C. A. Gold nanoparticles for biology and medicine. *Spherical Nucleic Acids*, pp. 55–90, 2020.

Goldsmith, B. R., Boley, M., Vreeken, J., Scheffler, M., and Ghiringhelli, L. M. Uncovering structure-property relationships of materials by subgroup discovery. *New Journal of Physics*, 19(1):013031, 2017.

Grosskreutz, H. and Rüping, S. On subgroup discovery in numerical domains. *Data Mining and Knowledge Discovery*, pp. 210–226, October 2009.

Helal, S. Subgroup discovery algorithms: a survey and empirical evaluation. *Journal of computer science and technology*, 31:561–576, 2016.

Hochreiter, S. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 6(02):107–116, 1998.

Kalofolias, J. and Vreeken, J. Naming the most anomalous cluster in hilbert space for structures with attribute information. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 2022.

Kalofolias, J., Boley, M., and Vreeken, J. Discovering robustly connected subgraphs with simple descriptions. In *2019 IEEE International Conference on Data Mining (ICDM)*, pp. 1150–1155, November 2019.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In Bengio, Y. and LeCun, Y. (eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

Klösgen, W. Explora: A multipattern and multistrategy discovery assistant. In *Advances in knowledge discovery and data mining*, pp. 249–271. 1996.

Lavrač, N., Kavšek, B., Flach, P., and Todorovski, L. Subgroup discovery with CN2-SD. *Journal of Machine Learning Research*, pp. 153–188, 2004.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Lemmerich, F. and Becker, M. pysubgroup: Easy-to-use subgroup discovery in python. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 658–662, 2018.

Ortiz, I. and Cummins, M. Global inequality: Beyond the bottom billion–a rapid review of income distribution in 141 countries. *UNICEF, Division of Policy and Strategy Working papers*, (1102), 2011.

Ou, Z., Xu, T., Su, Q., Li, Y., Zhao, P., and Bian, Y. Learning neural set functions under the optimal subset oracle. *Advances in Neural Information Processing Systems*, 35, 2022.

Papamakarios, G., Nalisnick, E., Rezende, D. J., Mohamed, S., and Lakshminarayanan, B. Normalizing flows for probabilistic modeling and inference. *The Journal of Machine Learning Research*, 22(1):2617–2680, 2021.

Proença, H. M., Grünwald, P., Bäck, T., and van Leeuwen, M. Robust subgroup discovery: Discovering subgroup lists using mdl. *Data Mining and Knowledge Discovery*, 36(5):1885–1970, 2022.

Qiao, L., Wang, W., and Lin, B. Learning accurate and interpretable decision rule sets from neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 4303–4311, 2021.

Rezende, D. and Mohamed, S. Variational inference with normalizing flows. In *International Conference on Machine Learning*, pp. 1530–1538. PMLR, 2015a.

Rezende, D. and Mohamed, S. Variational inference with normalizing flows. In *International conference on machine learning*, pp. 1530–1538. PMLR, 2015b.

Song, H., Kull, M., Flach, P., and Kalogridis, G. Subgroup Discovery with Proper Scoring Rules. In *Machine Learning and Knowledge Discovery in Databases*, pp. 492–510. Springer, Cham, September 2016.

Sutton, C., Boley, M., Ghiringhelli, L. M., Rupp, M., Vreeken, J., and Scheffler, M. Identifying domains of applicability of machine learning models for materials science. *Nature communications*, 11(1):4428, 2020.

van Leeuwen, M. and Knobbe, A. Non-redundant subgroup discovery in large and complex data. In *Machine Learning and Knowledge Discovery in Databases*, pp. 459–474. Springer, 2011.

Walter, N. P., Fischer, J., and Vreeken, J. Finding interpretable class-specific patterns through efficient neural search. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2024.

Wang, Z., Zhang, W., Liu, N., and Wang, J. Transparent Classification with Multilayer Logical Perceptrons and Random Binarization. *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 6331–6339, April 2020.

Wang, Z., Zhang, W., Liu, N., and Wang, J. Scalable rule-based representation learning for interpretable classification. *Advances in Neural Information Processing Systems*, 34:30479–30491, 2021.

Yang, Y., Morillo, I. G., and Hospedales, T. M. Deep neural decision trees. *arXiv preprint arXiv:1806.06988*, 2018.

Zaheer, M., Kottur, S., Ravanbakhsh, S., Poczos, B., Salakhutdinov, R. R., and Smola, A. J. Deep sets. *Advances in neural information processing systems*, 30, 2017.

A. Proof of Asymptotic Correctness of Soft-Binning

Proof: *Consider a real value $x_i \in \mathbb{R}$ and M sorted bin thresholds $\beta_{i,j} \in \mathbb{R}$, i.e. $\beta_{i,j} < \beta_{i,j+1}$. From the thresholds $\beta_{i,j}$, we construct the bias vector $b_i \in \mathbb{R}^{M+1}$ defined as*

$$b_i = (0, \sum_{j=1}^{1} -\beta_{i,j}, \ldots, \sum_{j=1}^{M} -\beta_{i,j})^T \ .$$

Additionally, we define a weight vector $w \in \mathbb{R}^{M+1}$ with $w_j = j$, so that

$$w = (1, 2, \ldots, M+1)^T \ .$$

The soft-binning result $z \in [0,1]^{M+1}$ is defined as

$$z = \textit{softmax}\left((wx_i + b_i)/t\right) \ .$$

Now, let x_i be in the l-th bin, i.e. $\beta_{i,l-1} < x_i < \beta_{i,l}$, then we now firstly prove that $\forall j \neq l : z_l > z_j$. We do this by showing that the l-th logit $\bar{z}_l = w_l x_i + b_{i,l}$ is the largest and hence also has the highest softmax activation.

Firstly, note that the bin thresholds are sorted in order, so that for $j < l$ it also holds that $\beta_{i,j} < \beta_{i,l}$. $\bar{z}_l$ is defined as

$$\bar{z}_l = w_l x_i + b_{i,l} = w_l x_i - \sum_{k=1}^{l-1} \beta_{i,k} \ .$$

We can simply transform $\bar{z}_l$ into $\bar{z}_{l-1}$ by subtracting $x_i - \beta_{i,l-1}$, so that

$$\bar{z}_l - x_i + \beta_{i,l} = w_{l-1} x_i - \sum_{k=1}^{l-2} \beta_{i,k} = \bar{z}_{l-1} \ .$$

Now, as x_i is in the l-th bin, we know that $\beta_{i,l-1} < x_i$ and hence $x_i - \beta_{i,l} < 0$. For all other $j < l$ $\beta_{i,j} < x_i$ holds, and hence also $\bar{z}_l > \bar{z}_j$.

Now consider the case where $j > l$. Here, it holds that

$$\bar{z}_l + x_i - \beta_{i,l+1} = w_{l+1} x_i - \sum_{k=1}^{l+1} \beta_{i,k} = \bar{z}_{l+1} \ .$$

In general, we may transform $\bar{z}_l$ into $\bar{z}_j$ by repeatedly adding $x_i - \beta_{i,k}$ for $k \in [l+1, \ldots, j]$. For all thresholds $x_i < \beta_{i,k}$ holds. Thus, each time we add a strictly negative number to the logit $\bar{z}_l$, which proves that also here $\forall j > l : \bar{z}_l > \bar{z}_j$. Thus, it holds that $\forall j \neq l : \bar{z}_l > \bar{z}_j$

Lastly, it remains to prove that with temperature $t \to 0$, z is a one-hot bin encoding, i.e. $z_l = 1$ and $\forall j \neq l : z_j = 0$. The soft-binning of z_l is defined as

$$\lim_{t \to 0} z_l = \lim_{t \to 0} \frac{\exp(\bar{z}_l/t)}{\sum_{j=1}^{M+1} \exp(\bar{z}_j/t)} = \lim_{t \to 0} \frac{1}{\sum_{j=1}^{M+1} \exp\left((\bar{z}_j - \bar{z}_l)/t\right)} \ .$$

For $j = l$, the sum term evaluates to $\exp(\bar{z}_l - \bar{z}_l)/t = \exp(0) = 1$. For $j \neq l$, it holds that $\bar{z}_l > \bar{z}_j$ as show previously, and hence in the limit

$$\lim_{t \to 0} \exp(\bar{z}_j - \bar{z}_j)/t = \exp(-\infty) = 0 \ .$$

Thus, in the limit $t \to 0$, the denominator sums up to 1 and hence $z_l = 1$, and as the softmax is positive and sums up to zero, it follows that $\forall j \neq l : z_j = 0$. □

B. Proof of Theorem 2

Theorem 2 *When Eqs. (5), (6), (7), and (8) hold, it is*

$$p_{Y|S=1}(y) - \int_{\mathbf{x}\in\mathbb{R}^m_{\epsilon}} p_{Y|S=1,\mathbf{X}}(y,\mathbf{x})d\mathbf{x} \leq \frac{C_Y(\epsilon_2 + C_{\mathbf{X}}\epsilon_1)}{\mathbb{P}(S=1)}\,.$$

Further, during learning, this bound becomes tighter until it asymptotically vanishes, assuming a decreasing annealing schedule for the temperature parameter.

Proof: *We first recall that, under our model, $p_{S=1|\mathbf{X}}(\mathbf{x}) = s_{t\to 0}(\mathbf{x};\alpha,\beta)$ for some $\alpha,\beta\in\mathbb{R}^n$, and is therefore a smooth function of $\mathbf{x}$. Intuitively, there are two regions of interest within $\mathbb{R}^m_{\notin}$: one within which it transitions from the value of almost 1 to that of almost 0, which is the region $\mathbb{R}^m_{\notin}\setminus\mathbb{R}^m_{\notin\subset}$, and a saturation region, where $p_{S=1|\mathbf{x}}\to 0$ super-exponentially, which is the region $\mathbb{R}^m_{\notin\subset}$. The particular thresholds that define these regions are not important, and any reasonable scheme leads to vanishing bounds ϵ_1, ϵ_2.*

More formally, using the partitioning of $\mathbb{R}^n$, we can split the integral of Eq. (4) into

$$p_{Y|S=1}(y) = \int_{\mathbf{x}\in\mathbb{R}^m_{\epsilon}} p_{Y|S=1,\mathbf{X}}(y,\mathbf{x})\frac{p_{S=1|\mathbf{X}}(\mathbf{x})p_{\mathbf{X}}(\mathbf{x})}{\mathbb{P}(S=1)}dx + \int_{\mathbf{x}\in\mathbb{R}^m_{\notin}} p_{Y|S=1,\mathbf{X}}(y,\mathbf{x})\frac{p_{S=1|\mathbf{X}}(\mathbf{x})p_{\mathbf{X}}(\mathbf{x})}{\mathbb{P}(S=1)}dx\,,$$

with the goal to upper bound (and hence ignore) the second term, which we consider as an error. This second term can be now bounded as

$$\begin{aligned}
&\int_{\mathbf{x}\in\mathbb{R}^m_{\notin}} p_{Y|S=1,\mathbf{X}}(y,\mathbf{x})\frac{p_{S=1|\mathbf{X}}(\mathbf{x})p_{\mathbf{X}}(\mathbf{x})}{\mathbb{P}(S=1)}d\mathbf{x} \leq \\
&\qquad \frac{1}{\mathbb{P}(S=1)}\int_{\mathbf{x}\in\mathbb{R}^m_{\notin}} C_Y p_{S=1|\mathbf{X}}(\mathbf{x})p_{\mathbf{X}}(\mathbf{x})d\mathbf{x} \leq \\
&\qquad \frac{C_Y}{\mathbb{P}(S=1)}\left[\int_{\mathbf{x}\in\mathbb{R}^m_{\notin}\setminus\mathbb{R}^m_{\notin\subset}} \underbrace{p_{S=1|\mathbf{X}}(\mathbf{x})}_{\leq 1} p_{\mathbf{X}}(\mathbf{x})d\mathbf{x} + \int_{\mathbf{x}\in\mathbb{R}^m_{\notin\subset}} p_{S=1|\mathbf{X}}(\mathbf{x})\underbrace{p_{\mathbf{X}}(\mathbf{x})}_{\leq C_{\mathbf{X}}}d\mathbf{x}\right] \leq \\
&\qquad \frac{C_Y}{\mathbb{P}(S=1)}\left[\int_{\mathbf{x}\in\mathbb{R}^m_{\notin}\setminus\mathbb{R}^m_{\notin\subset}} p_{\mathbf{X}}(\mathbf{x})d\mathbf{x} + C_{\mathbf{X}}\int_{\mathbf{x}\in\mathbb{R}^m_{\notin\subset}} p_{S=1|\mathbf{X}}(\mathbf{x})d\mathbf{x}\right] \leq \\
&\qquad\qquad \frac{C_Y(\epsilon_2 + C_{\mathbf{X}}\epsilon_1)}{\mathbb{P}(S=1)}\,,
\end{aligned}$$

where $p_{S=1|\mathbf{X}} \leq 1$ since S is a discrete random variable.

We argue about the second part by claiming that both bounds ϵ_1 and ϵ_2 vanish during learning. Indeed, the form $s(\mathbf{x}) \to p_{S=1|\mathbf{X}}(\mathbf{x})$ satisfies the assumption of Eq. (7) for a steep enough temperature parameter t, while it is also learning the correct domain $\mathbb{R}^m_{\notin}$, so that indeed the assumption of Eq. (8) is satisfied, both with inexorably diminishing bounds ϵ_1 and ϵ_2, respectively. □

C. Algorithm and Hyperparameters

In this section, we provide pseudocode for SYFLOW.

Algorithm 1: fit_flow($\{x^{(1)}, \ldots, x^{(n)}\}$,$\{y^{(1)}, \ldots, y^{(n)}\}$,$s$,$p$,$\bar{p}$)

$\log \mathcal{L} \leftarrow \frac{1}{n} \sum_{i=1}^{n} \log[p(y^{(i)}) \cdot s(x^{(i)}) + \bar{p}(y^{(i)}) \cdot (1 - s(x^{(i)}))]$;
loss $\leftarrow -\log \mathcal{L}$;
loss.backwards();
Update p;

Algorithm 2: SYFLOW(X, Y, epochs$_{\text{Flow}_Y}$, epochs$_{\text{Flow}_{Y_s}}$, lr$_{\text{Flow}}$,lr$_s$,priors, γ, λ, t)

$p_Y \leftarrow$ Neural Spline Flow;
for $i \leftarrow 1$ **to** *epochs*$_{Flow_Y}$ **do**
    $\log \mathcal{L} \leftarrow \frac{1}{n} \sum_{i=1}^{n} \log[p_Y(y^{(i)})]$;
    loss $\leftarrow -\log \mathcal{L}$;
    loss.backwards();
    Update p_Y;
end
$\alpha_i \leftarrow \min X_i$;
$\beta_i \leftarrow \max X_i$;
$a_i \leftarrow 1$;
Rule$(x) \leftarrow s(x; \alpha, \beta, a, t)$;
$p_{Y|S=1} \leftarrow$ Neural Spline Flow;
$p_{Y|S=0} \leftarrow$ Neural Spline Flow;
for $i \leftarrow 1$ **to** *epochs*$_{Flow_Y}$ **do**
    Compute subgroup membership probabilities $s(x^{(i)})$;
    KL $\leftarrow \frac{1}{n} \sum_{i=1}^{n} s(x^{(i)}) \cdot (\log[p_{Y|S=1}(y^{(i)})] - \log[p_Y(y^{(i)})])$;
    $n_s \leftarrow \frac{1}{n} \sum_{i=1}^{n} s(x^{(i)})$;
    Weighted-$KL \leftarrow KL \cdot \bar{s}^{\gamma}$;
    Regularizer $\leftarrow 0$;
    for p_{SG_k} ***in*** *priors* **do**
        Regularizer $\leftarrow$ Regularizer + $\sum_{i=1}^{n} s(x^{(i)}) \cdot (\log[p_{Y|S=1}(y^{(i)})] - \log[p_{Y|SG_k=1}(y^{(i)})])$;
    end
    Regularizer$\leftarrow \frac{\lambda}{|priors|}$·Regularizer ;
    loss $\leftarrow -$ Weighted-KL $-$Regularizer;
    loss.backwards();
    Update rule s to minimize objective/maximize weighted, regularized KL;
    fit_flow($\{x^{(1)}, \ldots, x^{(n)}\}$,$\{y^{(1)}, \ldots, y^{(n)}\}$,$s$,$p_{Y|S=1}$,$p_{Y|S=0}$);
    if $(t = \textit{epochs}_{Flow_{Y_s}}/2) \vee (t = \textit{epochs}_{Flow_{Y_s}} \cdot 3/4)$ **then**
        $t \leftarrow t/2$;
    end
end
return Rule, $p_{Y|S=1}$;

D. Hyperparameters for experimental evaluation

For all methods we optimize their respective hyperparameters such to maximize the measures used to evaluate the experiments. Since, BH has no hyperparameters no fine-tuning is required.

D.1. Synthetic experiments

We used one hyperparameter setting for all synthetic experiments For SYFLOW the hyperparameter setting is: $t = 0.2$, $\gamma = 0.5$, $\lambda = 0.5$, $\text{lr}_{\text{Flow}} = 5 \times 10^{-2}$, $\text{lr}_{\text{s}} = 2 \times 10^{-2}$, $\text{epochs}_{\text{Flow}_{\text{Y}}} = 2000$ and $\text{epochs}_{\text{Flow}_{\text{Y}_\text{s}}} = 1500$. For SD-$\mu$, SD-KL and RSD, we used 20 cutpoints and a beamwidth of 100, while γ is set to 1.0. Although γ has for all approaches the same functionality i.e. control the size of the subgroup, the absolute value of a γ has drastically different meanings. As SYFLOW outputs a soft assignment, we require a smaller alpha to achieve the same effect. For example, if SYFLOW assigns 0.9, then contribution to the size correction for $\gamma = 0.5$ is $\sqrt{0.9} \approx 0.95$, thus we require a smaller γ.

D.2. Real world data

For the experiments on Kaggle and UCI data (i.e. Section 5.2), we used for SYFLOW: $t = 0.2$, $\gamma = 0.3$, $\lambda = 2.0$, $\text{lr}_{\text{Flow}} = 5 \times 10^{-2}$, $\text{lr}_{\text{s}} = 2 \times 10^{-2}$, $\text{epochs}_{\text{Flow}_{\text{Y}}} = 1000$ and $\text{epochs}_{\text{Flow}_{\text{Y}_\text{s}}} = 1000$. For SD-$\mu$, SD-KL and RSD, we used 20 cutpoints, a beamwidth of 100, and while $\gamma = 1.0$. For the experiments on gold cluster data present in Figure 1c, we used $\text{epochs}_{\text{Flow}_{\text{Y}}} = 7000$, $\text{epochs}_{\text{Flow}_{\text{Y}_\text{s}}} = 3000$, $\lambda = 10.0$ and $\gamma = 0.2$ For the case study, we used $\text{epochs}_{\text{Flow}_{\text{Y}}} = 3000$, $\text{epochs}_{\text{Flow}_{\text{Y}_\text{s}}} = 2000$, $\lambda = 5$ and $\gamma = 0.2$.

E. Evaluation metric

To objectively evaluate the discovered subgroups on real-world data, we use Bhattacharyya coefficient (BC), KL-divergence (KL) and absolute mean distance (AMD) between the distribution of the subgroup $P_{Y|S=1}$ and overall distribution of P_Y. KL and AMD are size corrected, since both metrics are heavily influenced by the size of the subgroup. We use histograms to estimate the probability distributions. The edges of the bins are computed using the Freedman Diaconis Estimator, which is robust to outliers and less sensitive towards distribution shapes. For the subgroup distribution $P_{Y|S=1}$ and overall distribution P_Y, the metrics are formally defined as

$$BC(P_{Y|S=1}, P_Y) = \sum_{y \in \mathcal{Y}} \sqrt{p_{Y|S=1}(y) p_Y(y)}$$

$$D_{KL}(P_{Y|S=1}, P_Y) = \sum_{y \in \mathcal{Y}} p_{Y|S=1}(y) \log\left(\frac{p_{Y|S=1}(y)}{p_Y(y)}\right)$$

$$AMD(\mathcal{Y}_s, \mathcal{Y}) = \left| \left(\frac{1}{|\mathcal{Y}_s|} \sum_{y \in \mathcal{Y}_s} y \right) - \left(\frac{1}{|\mathcal{Y}|} \sum_{y \in \mathcal{Y}} y \right) \right| .$$

In the last definition, with slight abuse of notation, we used $\mathcal{Y}_s$ to denote all points in the subgroup. For the Table 1, we size corrected KL and AMD using $\gamma = 1$. Note, this is exactly the metric that SD-μ and SD-KL optimize.

F. Rule Complexity Experiment

We study how SYFLOW learns subgroups of increasing complexity. This is achieved by increasing the number of predicates in a generated rule up to ten. We keep the target distribution fixed to a Normal distribution.

We present our results in Figure 7; Here, as the complexity of the rule increases, the accuracy of all methods generally decreases as the task becomes progressively harder. Amongst all methods, SYFLOW recovers the planted subgroup with the highest accuracy. In particular, SYFLOW improves the most over its competitors on the more complex subgroups.

G. Target distributions for synthetic experiment

In Figure 8, we show the different target distributions for the first synthetic experiment.

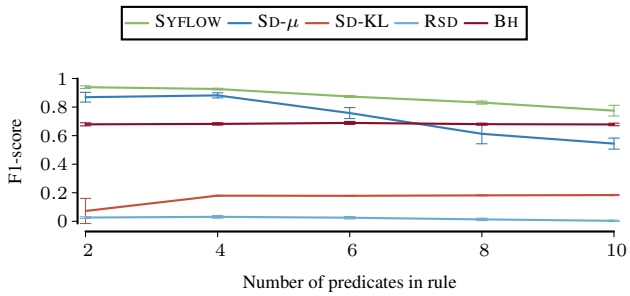
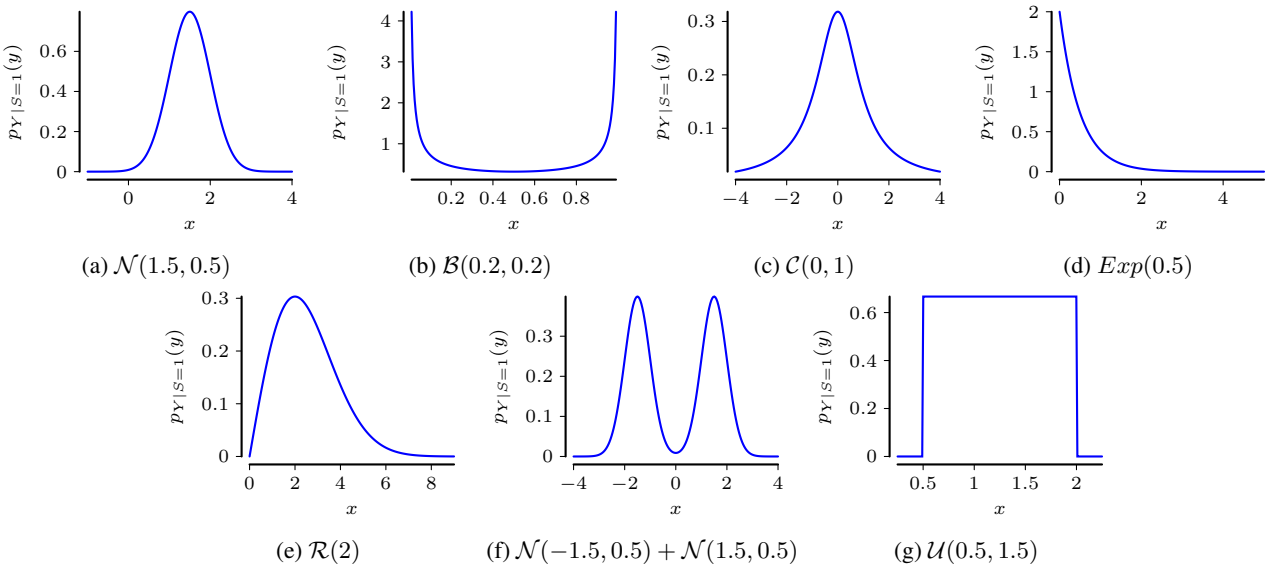


Figure 7. F1 under increasingly complex rules, higher is better



(a) $\mathcal{N}(1.5, 0.5)$ (b) $\mathcal{B}(0.2, 0.2)$ (c) $\mathcal{C}(0, 1)$ (d) $Exp(0.5)$

(e) $\mathcal{R}(2)$ (f) $\mathcal{N}(-1.5, 0.5) + \mathcal{N}(1.5, 0.5)$ (g) $\mathcal{U}(0.5, 1.5)$

Figure 8. Here, we show the different target distributions for second synthetic experiment in Section 5.

H. Subgroup descriptions

We show in Table 2 further examples of subgroups found on the real life datasets. For each method we select the first subgroup that the respective method found. Since BH did not find relevant subgroups for most datasets, we do not consider it in the table.

I. Subgroup Discovery on Image Data

One of the defining characteristics of SYFLOW is its ability to give an interpretable rule for each subgroup. Hence, we focus our experiments on tabular data where we can directly interpret the learned rules. Still, SYFLOW also has potential applications with structured data such as images. We conduct a preliminary experiment on the MNIST dataset (LeCun et al., 1998) using the digits 0 and 1 only. As the target variable Y, we use pixel values of the downsampled, normalized image in 14×14 resolution as a vector $Y \in \mathbb{R}^{196}$. For the feature variable X, preprocessing is required for the rule learner because of the high variability between individual pixel values. Instead, we use the two-dimensional t-SNE embedding of the pixel values as features $X \in \mathbb{R}^2$.

We show in Figure 9 the discovered subgroups for the digits 0 and 1. SYFLOW is able to learn a subgroup of resp. the digit 0 and the digit 1, and the sampled images from the subgroup distributions closely resemble the digits. The corresponding descriptors are boxes around the regions of the t-SNE embedding where the digit is located. This shows that in principle, SYFLOW can be applied to image data. Future work could focus on extending SYFLOW to directly work with image data, for example by using convolutional neural networks to learn the subgroups and giving up the interpretability of the rules, or

Dataset	Method	Rule
Abalone	SYFLOW	$0.20 < \text{whole-weight} < 2.83 \wedge 0.13 < \text{viscera-weight} < 0.60 \wedge 0.26 < \text{shell-weight} < 1.00 \wedge \text{sex-I} = 0$
	RSD	$\text{shell-weight} \geq 0.43 \wedge 0.46 \leq \text{diameter} < 0.49 \wedge \text{shucked-weight} < 0.55 \wedge \text{whole-weight} \geq 1.03$
	SD-KL	$\text{shell-weight} < 0.48$
	SD-μ	$\text{height} \geq 0.10 \wedge \text{shucked-weight} \geq 0.16 \wedge \text{shell-weight} \geq 0.11 \wedge \text{shell-weight} \geq 0.19$
Airquality	SYFLOW	$404.67 < \text{NOx(GT)} < 1479.00 \wedge -44.90 < \text{NO2(GT)} < 340.00$
	RSD	$\text{PT08.S4(NO2)} \geq 2026.0 \wedge \text{PT08.S3(NOx)} < 373.0 \wedge \text{month} \geq 11.0$
	SD-KL	$\text{PT08.S1(CO)} < 1285.0 \wedge \text{NMHC(GT)} < 185.0 \wedge \text{PT08.S2(NMHC)} < 1068.0 \wedge \text{NOx(GT)} < 326.0 \wedge \text{NOx(GT)} \geq -200.0 \wedge \text{PT08.S5(O3)} < 1780.0$
	SD-μ	$\text{PT08.S1(CO)} \geq 956.0 \wedge \text{PT08.S2(NMHC)} \geq 979.0 \wedge \text{PT08.S5(O3)} \geq 704.0 \wedge \text{PT08.S5(O3)} \geq 917.0$
Automobile	SYFLOW	$\text{If } 159.42 < \text{length} < 208.10 \wedge 60.64 < \text{width} < 72.00 \wedge 2684.19 < \text{curb-weight} < 4066.00 \wedge 64.84 < \text{engine-size} < 270.46 \wedge 7.00 < \text{compression-ratio} < 19.11 \wedge 16.00 < \text{highway-mpg} < 31.72 \wedge \text{jaguar}=0 \wedge \text{mercedes-benz}=0 \wedge \text{plymouth}=0 \wedge \text{subaru}=0 \wedge \text{fuel-type-gas}=1 \wedge \text{engine-location-front}=1 \wedge \text{engine-location-rear}=0 \wedge \text{engine-type-ohcf}=0 \wedge \text{fuel-system-2bbl}=0 \wedge \text{fuel-system-mpfi}=1$
	RSD	$\text{engine-size} \geq 201.5$
	SD-KL	$\text{highway-mpg} \geq 29.0 \wedge \text{audi}=0 \wedge \text{bmw}=0 \wedge \text{wheel-base} < 104.90 \wedge \text{width} < 66.90 \wedge \text{body-style-convertible}=0 \wedge \text{engine-size} < 161.0$
	SD-μ	$\text{highway-mpg} < 30.0 \wedge \text{honda}=0 \wedge \text{isuzu}=0 \wedge \text{plymouth}=0 \wedge \text{subaru}=0 \wedge \text{curb-weight} \geq 2385.0 \wedge \text{fuel-system-mfi}=0$
Bike	SYFLOW	$\text{holiday}=0 \wedge 1.00 < \text{weathersit} < 2.35 \wedge 0.47 < \text{atemp} < 0.78 \wedge 0.18 < \text{hum} < 0.83 \wedge \text{season-1}=0$
	RSD	$\text{temp} < 0.2$
	SD-KL	$\text{temp} \geq 0.26 \wedge \text{atemp} \geq 0.28 \wedge \text{hum} < 0.87 \wedge \text{windspeed} < 0.34 \wedge \text{season-1}=0$
	SD-μ	$\text{temp} \geq 0.40 \wedge \text{temp} \geq 0.43 \wedge \text{hum} < 0.82$
California	SYFLOW	$0.50 < \text{MedInc} < 2.4$
	RSD	$\text{MedInc} \geq 7.4 \wedge 35.0 \leq \text{HouseAge} < 38.0 \wedge -121.19 \leq \text{Longitude} < -118.34 \wedge \text{AveBedrms} \geq 0.96$
	SD-KL	$\text{MedInc} < 5.11 \wedge \text{MedInc} < 6.16 \wedge \text{HouseAge} < 52.0 \wedge \text{AveOccup} \geq 2.08$
	SD-μ	$\text{MedInc} \geq 3.32 \wedge \text{AveOccup} < 3.89 \wedge \text{AveOccup} < 4.33 \wedge \text{Latitude} < 37.99 \wedge \text{Longitude} < -117.08$
Insurance	SYFLOW	$44.00 < \text{age} < 64.00 \wedge \text{smoker}=0$
	RSD	$\text{smoker}=1 \wedge \text{bmi} \geq 30.0 \wedge \text{age} \geq 59.0$
	SD-KL	$\text{smoker}=0$
	SD-μ	$\text{smoker}=1$
Mpg	SYFLOW	$3.35 < \text{cylinders} < 5.25$
	RSD	$\text{weight} \geq 3845.0 \wedge 71.0 \leq \text{model-year} < 74.5$
	SD-KL	$\text{displacement} \geq 151.0 \wedge \text{weight} \geq 2671.0 \wedge \text{weight} \geq 3085.0$
	SD-μ	$\text{cylinders}=4.0 \wedge \text{weight} < 2807.0 \wedge \text{weight} < 4278.0$
Student	SYFLOW	$\text{school}=1 \wedge \text{address}=0 \wedge \text{failures}=0 \wedge \text{schoolsup}=1 \wedge \text{nursery}=0 \wedge \text{higher}=0 \wedge \text{internet}=1 \wedge 1.00 < \text{Dalc} < 3.34 \wedge \text{Medu-1}=0 \wedge \text{Fedu-1}=0$
	RSD	$\text{failures} = 1 \wedge \text{famsize}=1 \wedge \text{absences} < 1 \wedge \text{famsup}=0$
	SD-KL	$\text{failures}=0$
	SD-μ	$\text{school}=0 \wedge \text{higher}=0 \wedge \text{absences} < 16 \wedge \text{absences} < 18 \wedge \text{failures}=0 \wedge \text{schoolsup}=1$
Wages	SYFLOW	$\text{sex}=0 \wedge 10.68 < \text{ed} < 18.00 \wedge 29.86 < \text{age} < 95.00$
	RSD	$\text{ed} \geq 18.0 \wedge 37.0 \leq \text{age} < 43.0 \wedge \text{sex}=0 \wedge \text{height} < 70.25$
	SD-KL	$\text{race-other}=0.0$
	SD-μ	$\text{height} \geq 65.05 \wedge \text{height} \geq 65.79 \wedge \text{sex}=0 \wedge \text{age} \geq 30.0$
Wine	SYFLOW	$0.23 < \text{volatile acidity} < 1.10 \wedge 0.99 < \text{density} < 1.04 \wedge 8.00 < \text{alcohol} < 10.12$
	RSD	$\text{alcohol} \geq 12.75 \wedge 29.0 < \text{free sulfur dioxide} < 41.0 \wedge \text{pH} < 3.24 \wedge 0.26 \leq \text{citric acid¡0.34} \wedge \text{residual sugar} \geq 1.4$
	SD-KL	$\text{free sulfur dioxide} \geq 11.0$
	SD-μ	$\text{fixed acidity} < 8.30 \wedge \text{alcohol} \geq 10.40 \wedge \text{free sulfur dioxide} \geq 11.0 \wedge \text{total sulfur dioxide} < 195.0 \wedge \text{density} < 1.00$

Table 2. Symbolic subgroup descriptions for real life datasets in Section 5.2

Table 3. We show the standard deviation for the 100 reruns of SYFLOW on real world data.

Dataset	D_{KL}	BC	AMD
abalone	0.01	0.02	0.04
airquality	0.01	0.02	0.02
automobile	0.02	0.04	239.14
bike	0.0	0.01	73.98
california	0.01	0.04	0.02
insurance	0.01	0.1	320.22
mpg	0.0	0.0	0.08
student	0.02	0.05	0.1
wages	0.0	0.02	172.48
wine	0.01	0.03	0.03

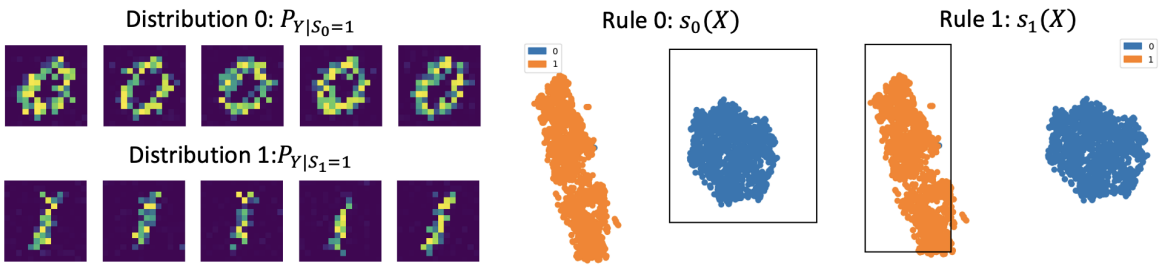


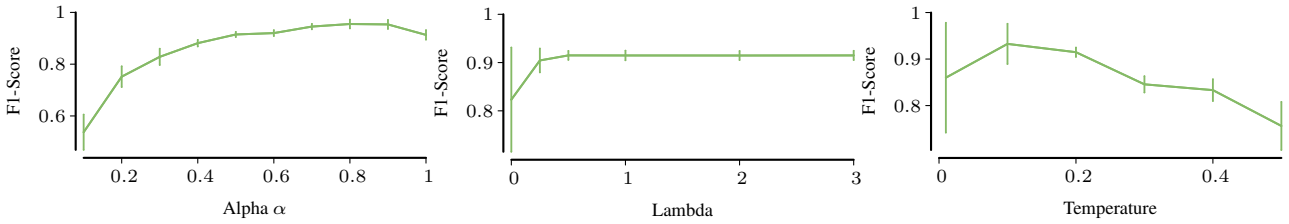
Figure 9. Subgroup discovery on the MNIST dataset with features X obtained by 2D t-SNE embedding. The discovered subgroups resemble the digits 0 and 1 resp.

by using an image-targeted normalizing flow architecture to model the subgroup distribution.

J. Sensitivity of SYFLOW

We here provide preliminary results for the average F1 score across the different target distributions, shown in Figure D. Overall, SYFLOW performs well with a variety of different hyperparameters, and gives the end user room for customization in regards to the desired size of the subgroups through γ and the diversity between subgroups through λ. When it comes to discovering underlying subgroup structures, as demonstrated in the experiments, SYFLOW does so reliably for a wide range of parameters.

In Table 3, we show the standard deviations of SYFLOW for the 100 runs on real world data.



(a) **Subgroup size weighting** γ (b) **Regularization strength** λ (c) **Predicate temperature/softness** t

Figure 10. Here we conduct a sensitivity analysis of the hyperparameters of SYFLOW. **Subgroup size weighting γ:** SYFLOW works well in a range of values for $\gamma \in [0.4, 1]$. **Regularization strength λ:** Using the regularizer improves the F1 score by 0.05, whilst varying its strength between $[0.5, 3]$ does not have a noticeable impact on accuracy. **Predicate temperature/softness t:** the starting temperature controls the softness of the binning function. Its ideal range by empirical observation lies in between $[0.01, 0.2]$. Setting it too high can result in un-crisp binning functions, which in turn results in an inaccurate subgroup assignment yielding a worse F1 score. In both synthetic and real-world experiments we used a temperature of $t = 0.2$ which provides good performance across the board