Global Explanations with Decision Rules: a Co-learning Approach

Géraldin Nanfack[1] Paul Temple[1] Benoît Frénay[1]

[1]PReCISE Research Center, Namur Digital Institute (NADI), University of Namur, Belgium

Abstract

Black-box machine learning models can be extremely accurate. Yet, in critical applications such as in healthcare or justice, if models cannot be explained, domain experts will be reluctant to use them. A common way to explain a black-box model is to approximate it by a simpler model such as a decision tree. In this paper, we propose a co-learning framework to learn decision rules as explanations of black-box models through knowledge distillation and simultaneously constrain the black-box model by these explanations; all of this in a differentiable manner. To do so, we introduce the soft truncated Gaussian mixture analysis (STruGMA), a probabilistic model which encapsulates hyper-rectangle decision rules. With STruGMA, global explanations can be extracted by any rule learner such as decision lists, sets or trees. We provide evidences through experiments that our framework can globally explain differentiable black-box models such as neural networks. In particular, the explanation fidelity is increased, while the accuracy of the models is marginally impacted.

1 INTRODUCTION

For more than a decade now, machine learning has been ever-increasingly applied to various fields. More recently, special emphasis has been put on the need for machine learning models to provide explanations for their predictions in human-understandable terms [Doshi-Velez and Kim, 2017, Ribeiro et al., 2016], in addition to accurate predictions. In domains, such as finance, justice and healthcare, if models fail to provide explanations, users may be reluctant to use them. In response, algorithms have been proposed to improve the performance of interpretable decision lists [Yang et al., 2017], sets [Mita et al., 2020] and trees [Verwer and Zhang, 2019]. However, in practice, more powerful models such as deep neural networks achieve impressive performances for tabular [Klambauer et al., 2017], image [Chen et al., 2020], and text [Devlin et al., 2019] data. These models are complex and do not provide embedded explanations, forcing users to rely on external tools to explain decisions.

Two main families of explanations can be used: *global explanations* which explain *entirely* a complex model on its whole input space; and *local explanations* where an explanation is valid only on a specific region, close to a particular instance [Guidotti et al., 2018]. This paper focuses on global explanations of black-box models using decision rules (if-then rules), which are the most famous non-linear form of explanations [Lundberg et al., 2020].

Existing approaches for global explanations with decision rules (if-then rules) show some issues. For instance, the theoretical formalisation of *post-hoc* methods [Craven and Shavlik, 1995, Ribeiro et al., 2018, Pedreschi et al., 2019, Confalonieri et al., 2020] is unclear [Craven and Shavlik, 1996, Wolf et al., 2019] as well as whether their explanations reflect accurately the black-box model [Kim et al., 2018, Slack et al., 2020]. Another approach is to constrain the black-box model to be easily explainable, through regularisation or optimisation for explainability. While few works focus on decision rules explanations, since decision rules [Okajima and Sadamasa,2019] and local regions [Wu et al., 2020] are supposed to be known a priori, we propose to derive decision rules directly from the black-box model.

This paper proposes to jointly learn the black-box model and its decision rules-based explanation. To bridge the gap between state-of-the-art rule learners [Verwer and Zhang, 2019, Mita et al., 2020] that involve discrete optimisation and differentiable black-box models, we introduce the soft truncated Gaussian mixture analysis (STruGMA), a probabilistic model that encapsulates learnable hyper-rectangle decision rules sets learnt via gradient descent. A co-learning framework is proposed to learn STruGMA through knowledge distillation. Simultaneously, the black-box model is constrained to reflect the hyper-rectangle decision rules explanations given by STruGMA.

The remainder of this paper is as follows. Section 2 presents related work. Section 3 provides details of the proposed STruGMA and our co-learning framework. Section 4 presents the evaluation protocol as well as results and discussion. Section 5 concludes with future work directions.

2 RELATED WORK

To globally explain black-box models with rules, existing methods either are post-hoc either use regularisation.

2.1 POST-HOC EXPLAINABILITY METHODS

Post-hoc explainability methods that use decision rules as explanations are generally called rule extraction methods. They consider a black-box model, and then learn an interpretable set, list or tree of decision rules to match its predictions. An early work is TREPAN [Craven and Shavlik, 1995], which approximates a neural network with a decision tree by learning m-of-n rules chosen to maximise the information gain ratio. There also exists a considerable literature of methods that use genetic algorithms [Boz, 2002, Arbatli and Akin, 1997], sampling strategies [Craven and Shavlik, 1994, Ribeiro et al., 2018] and convex predicates [Gopinath et al., 2019]. However, their main limitation is that they are not stable [Melis and Jaakkola, 2018]. In addition, there are no guarantees that explanations accurately reflect the knowledge captured by the complex black-box model [Kim et al., 2018, Slack et al., 2020].

2.2 REGULARISING FOR EXPLAINABILITY

Explaining black-box models with decision rules can also be done by regularising the black-box models. Two notable works are Okajima and Sadamasa [2019] and Wu et al. [2020]. The former proposes to change the neural network architecture such that it can predict a rule (from a predefined rule set) and then a label given a particular instance. The latter leverages Wu et al. [2018] to enforce explainability by decision trees in local regions known *a priori*. In addition to being a challenging task (because of the discrete nature of rules), regularising for rule explanations with predefined rule sets or local regions has a major prerequisite. These rule sets or local regions are assumed to be known *a priori*. For explainability purposes, this is impractical, since they should be derived from the black-box model.

To address this problem, we propose a co-learning framework where hyper-rectangle rules are embedded into the newly introduced soft-truncated Gaussian mixture analysis (STruGMA). During co-learning (see Figure 1), STruGMA tries to explain the black-box model by the use of knowledge distillation, while the black-box model is learned with a regularisation with respect to STruGMA. This framework shares similarities with mutual distillation and posterior regularisation [Zhang et al., 2018, Hu et al., 2016].

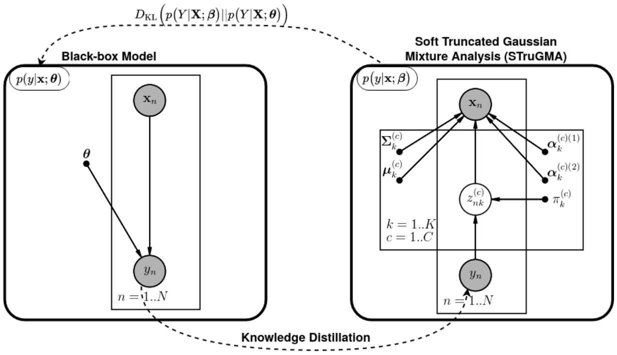

$D_{\mathrm{KL}}\left(p(Y|\mathbf{X};\boldsymbol{\beta})||p(Y|\mathbf{X};\boldsymbol{\theta})\right)$
Black-box Model
Soft Truncated Gaussian
Mixture Analysis (STruGMA)
$p(y|\mathbf{x};\boldsymbol{\theta})$
$\mathbf{x}_n$
$\boldsymbol{\theta}$
y_n
$n = 1..N$
$p(y|\mathbf{x};\boldsymbol{\beta})$
$\mathbf{x}_n$
$\boldsymbol{\Sigma}_k^{(c)}$
$\boldsymbol{\mu}_k^{(c)}$
$\boldsymbol{\alpha}_k^{(c)(1)}$
$\boldsymbol{\alpha}_k^{(c)(2)}$
$z_{nk}^{(c)}$
$\pi_k^{(c)}$
$k = 1..K$
$c = 1..C$
y_n
$n = 1..N$
Knowledge Distillation

Figure 1: Co-learning between the black-box model (left) and STruGMA (right) through knowledge distillation and regularisation (dashed lines).

3 ITERATIVE LEARNING OF RULES AND CONSTRAINED BLACK-BOX MODEL

To embed decision rules in a differentiable *surrogate*, Section 3.1 proposes the soft truncated Gaussian mixture analysis (STruGMA), Section 3.2 proposes solutions for challenges that arise when learning STruGMA and Section 3.3 presents our co-learning strategy with the black box model.

3.1 SOFT TRUNCATED GAUSSIAN MIXTURE ANALYSIS FOR DIFFERENTIABLE MODELLING

Geometrically, a rule defines a hyper-rectangle convex region $R(\boldsymbol{\alpha}_k) = \{\alpha_{kd}^{(1)} \leq x_d \leq \alpha_{kd}^{(2)}\}_{d=1}^{D}$, where $\alpha_{kd}^{(i)} \in \bar{\mathbb{R}}$ are the boundaries[1], D is the input space dimension and k is the index of the hyper-rectangle rule.

Motivated by the approximation properties of Gaussian distributions (thanks to the central limit theorem), we choose to map, as a surrogate, the k-th single rule to the truncated normal distribution

$$p\left(\boldsymbol{x}|z=k;\boldsymbol{\mu},\boldsymbol{\Sigma},\boldsymbol{\alpha}^{(1)},\boldsymbol{\alpha}^{(2)}\right) = \frac{\mathcal{N}(\boldsymbol{x};\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k)}{\int_{\boldsymbol{\alpha}_k^{(1)}}^{\boldsymbol{\alpha}_k^{(2)}} \mathcal{N}(\boldsymbol{t};\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k)d\boldsymbol{t}} \mathbb{1}\left\{\boldsymbol{\alpha}_k^{(1)} \leq \mathbf{x} \leq \boldsymbol{\alpha}_k^{(2)}\right\}(\boldsymbol{x}),$$

[1] Note that the upper boundary $\alpha_{kd}^{(2)}$ can be $+\infty$ and the lower boundary $\alpha_{kd}^{(1)}$ can be $-\infty$, whenever relevant.

where $\mathbb{1}\{.\}$ is the indicator function. Optimising such a distribution is numerically unstable because of the piecewise discontinuity of the indicator function $\mathbb{1}\{.\}$. However, the truncated normal distribution can be approximated by the soft truncated normal distribution [Souris et al., 2018]

$$p\left(\boldsymbol{x}|z=k;\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k,\boldsymbol{\alpha}^{(1)},\boldsymbol{\alpha}^{(2)}\right)\approx\frac{\mathcal{N}(\boldsymbol{x};\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k)}{\int_{\boldsymbol{\alpha}_k^{(1)}}^{\boldsymbol{\alpha}_k^{(2)}}\mathcal{N}(\boldsymbol{t};\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k)d\boldsymbol{t}}$$
$$\prod_{d=1}^{D}\sigma_\eta\left(x_d-\alpha_{kd}^{(1)}\right)\left(1-\sigma_\eta\left(x_d-\alpha_{kd}^{(2)}\right)\right),$$

where $\sigma_\eta(x)=1/(1+\exp(-\eta x))$ and η is a positive number. When $\eta\to+\infty,\sigma_\eta(x)$ tends towards $\mathbb{1}\{x\geq 0\}$. In practice, $\eta\geq 20$ is sufficient. Notice that this distribution reduces to the normal case when $\alpha_{kd}^{(1)}\to-\infty$ and $\alpha_{kd}^{(2)}\to+\infty$. Therefore, it can be interpreted as a normal distribution whose shape is constrained. Although its support is theoretically $\mathbb{R}$ in the univariate case, the high-density region is $[\alpha_{kd}^{(1)},\alpha_{kd}^{(2)}]$. Figure 2 shows an example for $\mathbb{R}^2$ with $\eta=20$.

Taking advantage of this distribution, we propose the (finite) soft truncated Gaussian mixture (STruGM) model to embed a set of hyper-rectangle rules in a differentiable model. In addition, by drawing inspiration from the mixture discriminant analysis (MDA) [Hastie and Tibshirani, 1996], we propose the soft truncated Gaussian mixture analysis (STruGMA), i.e., a probabilistic generative classifier with class-conditional STruGM distributions. In STruGMA, each class has its own STruGM. In other words, STruGMA is a classifier $p(y|\mathbf{x};\boldsymbol{\beta})$ that reduces, when conditioning on the class, to the class-specific STruGM $p(\mathbf{x}|y;\boldsymbol{\beta})=\sum_{k=1}^{K}p(z=k|y;\boldsymbol{\beta})p(\mathbf{x}|z=k,y;\boldsymbol{\beta})$, where K is the class-specific number of components and $\boldsymbol{\beta}$ are the parameters of STruGMA.

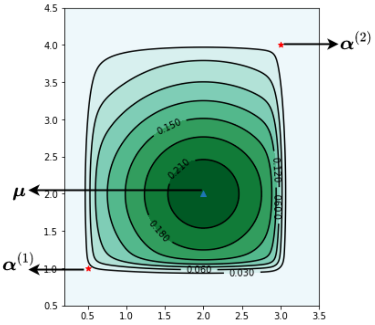


Figure 2: A soft truncated Normal Distribution.

3.2 ADAPTING EM FOR STRUGMA

Three challenges arise when learning STruGMA. Firstly, unlike the Gaussian distribution, it has been shown [Cohen Jr, 1950] that neither $\boldsymbol{\mu},\boldsymbol{\Sigma}$ nor $\boldsymbol{\alpha}$ have a closed-form solution for the maximum likelihood estimation (MLE) of a single truncated normal distribution. Secondly, learning the parameters $\boldsymbol{\alpha}^{(1)}$ and $\boldsymbol{\alpha}^{(2)}$ of STruGMA must satisfy the constraint $\boldsymbol{\alpha}^{(1)}<\boldsymbol{\alpha}^{(2)}$. Thirdly, learning STruGMA may result in many overlapping hyper-rectangle decision rules that are less interesting and more complex for explainability purposes [Fürnkranz et al., 2012, Lakkaraju et al., 2016].

STruGMA is a generative classifier with parameters $\boldsymbol{\beta}=\{\boldsymbol{\beta}^{(1)},...,\boldsymbol{\beta}^{(c)},...,\boldsymbol{\beta}^{(C)}\}$, where C is the number of classes and $\boldsymbol{\beta}^{(c)}=\{\boldsymbol{\pi}^{(c)},\boldsymbol{\mu}^{(c)},\boldsymbol{\alpha}^{(c)(1)},\boldsymbol{\alpha}^{(c)(2)},\boldsymbol{\Sigma}^{(c)}\}$. $\boldsymbol{\alpha}^{(c)(1)}$ (resp. $\boldsymbol{\alpha}^{(c)(2)}$) $\in\mathbb{R}^{K_c\times D}$ is the lower (resp. upper) truncated point of the k-th component of class c; similarly, $\boldsymbol{\mu}^{(c)}\in\mathbb{R}^{K_c\times D}$ and $\boldsymbol{\Sigma}^{(c)}\in\mathbb{R}^{K_c\times D\times D}$. $\boldsymbol{\pi}^{(c)}$ are the mixing parameters and K_c is the number of components of class c. Here, $\boldsymbol{\Sigma}_k^{(c)}$ is diagonal for the sake of factorisation of the denominator. It can be easily extended by computing the multivariate Gaussian cumulative distribution function. Given these parameters, the joint distribution of STruGMA is

$$p(\mathbf{x},y=c|\boldsymbol{\beta})=p(y=c)\sum_{k=1}^{K_c}p\left(z^{(c)}=k|y=c;\boldsymbol{\pi}_k^{(c)}\right)$$
$$p\left(\mathbf{x}|z^{(c)}=k,y=c;\boldsymbol{\mu}_k^{(c)},\boldsymbol{\Sigma}_k^{(c)},\boldsymbol{\alpha}_k^{(c)(1)},\boldsymbol{\alpha}_k^{(c)(2)}\right)$$
$$=p(y=c)\sum_{k=1}^{K_c}\boldsymbol{\pi}_k^{(c)}\frac{\mathcal{N}(\mathbf{x};\boldsymbol{\mu}_k^{(c)},\boldsymbol{\Sigma}_k^{(c)})}{\int_{\boldsymbol{\alpha}_k^{(c)(1)}}^{\boldsymbol{\alpha}_k^{(c)(2)}}\mathcal{N}(\mathbf{t};\boldsymbol{\mu}_k^{(c)},\boldsymbol{\Sigma}_k^{(c)})d\mathbf{t}}$$
$$\prod_{d=1}^{D}\sigma_\eta\left(x_d-\alpha_{kd}^{(c)(1)}\right)\left(1-\sigma_\eta\left(x_d-\alpha_{kd}^{(c)(2)}\right)\right).$$

From now on, for simplicity, the class conditioning c is omitted for parameters. One of the most popular method to learn finite mixture models is the expectation-maximisation (EM) algorithm. Therefore, as it consists of a mixture per class, STruGMA can be learned by adapting EM. With the parameters $\boldsymbol{\beta}=\{\boldsymbol{\pi},\boldsymbol{\mu},\boldsymbol{\Sigma},\boldsymbol{\alpha}^{(1)},\boldsymbol{\alpha}^{(2)}\}$, EM maximises the expected log-likelihood $Q(\boldsymbol{\beta},\boldsymbol{\beta}^t)$ by alternating the following steps:

- E-step: Computing class responsibilities

$$r_{nk}=p\left(z=k|\boldsymbol{x}_n;\boldsymbol{\beta}^t\right)=\frac{\pi_k p(\boldsymbol{x}_n|z=k;\boldsymbol{\beta}^t)}{\sum_{k_1}\pi_{k_1}p(\boldsymbol{x}_n|z=k_1;\boldsymbol{\beta}^t)};\tag{1}$$

- M-step: Because of the lack of closed-form solution of parameters through the MLE, the M-step performs a gradient descent on the negative expected loglikelihood

$$\boldsymbol{\beta}^{t+1}=\boldsymbol{\beta}^t+\epsilon_t\nabla Q\left(\boldsymbol{\beta},\boldsymbol{\beta}^t\right),\tag{2}$$

where ϵ_t is the learning rate and

$$
\begin{aligned}
Q(\boldsymbol{\beta}, \boldsymbol{\beta}^t) = \sum_n \sum_k r_{nk} \log \pi_k + \sum_n \sum_k r_{ik} \\
\Bigg[\log \mathcal{N}(\boldsymbol{x}_n; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) + \sum_d \log \sigma_\eta \left(x_{nd} - \alpha^{(1)}_{kd} \right) \\
+ \log \left(1 - \sigma_\eta \left(x_{nd} - \alpha^{(2)}_{kd} \right) \right) \Bigg] \\
- \sum_n \sum_k r_{nk} \log \int_{\boldsymbol{\alpha}_k^{(1)}}^{\boldsymbol{\alpha}_k^{(2)}} \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) d\mathbf{x}.
\end{aligned}
$$

Details about gradients $\nabla Q(\boldsymbol{\beta}, \boldsymbol{\beta}^t)$ are given in the supplementary material.

Directly optimising with gradient descent in the M-Step, as described, may raise difficulties caused by the definition of the denominator of the soft truncated normal distribution. Indeed, we need to impose $\boldsymbol{\alpha}^{(2)} > \boldsymbol{\alpha}^{(1)}$ as a hard constraint. To solve the problem, we leverage the projected gradient descent method [Boyd et al., 2004] on the constraint set $S = \{\boldsymbol{\alpha}^2 > \boldsymbol{\alpha}^1\}$. The projected gradient solves the problem

$$
\begin{aligned}
\boldsymbol{\alpha}^{t+1} =& \text{Proj}_{\boldsymbol{\alpha} \in S}(\boldsymbol{\alpha}^t + \epsilon_t \nabla Q(\boldsymbol{\alpha}, \boldsymbol{\alpha}^t)) \\
=& \text{argmin}_{\boldsymbol{\alpha}} || \boldsymbol{\alpha} - (\boldsymbol{\alpha}^t + \epsilon_t \nabla Q(\boldsymbol{\alpha}, \boldsymbol{\alpha}^t)) || \\
& \text{s.t. } \boldsymbol{\alpha}^{(2)} > \boldsymbol{\alpha}^{(1)}
\end{aligned}
$$

and, using the method of Lagrange multipliers on this constrained quadratic optimisation problem, one obtains

$$
\begin{cases}
\boldsymbol{\alpha}^{(1)}_{t+1} = \boldsymbol{\alpha}^{(1)}_t + \epsilon_t \nabla Q(\boldsymbol{\alpha}^{(1)}, \boldsymbol{\alpha}^{(1)}_t) \\
\boldsymbol{\alpha}^{(2)}_{t+1} = \boldsymbol{\alpha}^{(2)}_t + \epsilon_t \nabla Q(\boldsymbol{\alpha}^{(2)}, \boldsymbol{\alpha}^{(2)}_t)
\end{cases}
$$

if $\boldsymbol{\alpha}_{t+1} \in S$ and, otherwise,

$$
\begin{cases}
\boldsymbol{\alpha}^{(1)}_{t+1} = \frac{1}{2} \Big(\boldsymbol{\alpha}^{(1)}_t + \boldsymbol{\alpha}^{(2)}_t \\
\qquad + \epsilon_t \left(\nabla Q(\boldsymbol{\alpha}^{(1)}, \boldsymbol{\alpha}^{(1)}_t) + \nabla Q(\boldsymbol{\alpha}^{(2)}, \boldsymbol{\alpha}^{(2)}_t) \right) - \zeta \Big) \\
\boldsymbol{\alpha}^{(2)}_{t+1} = \frac{1}{2} \Big(\boldsymbol{\alpha}^{(1)}_t + \boldsymbol{\alpha}^{(2)}_t \\
\qquad + \epsilon_t \left(\nabla Q(\boldsymbol{\alpha}^{(1)}, \boldsymbol{\alpha}^{(1)}_t) + \nabla Q(\boldsymbol{\alpha}^{(2)}, \boldsymbol{\alpha}^{(2)}_t) \right) + \zeta \Big)
\end{cases}
$$

The margin $\zeta > 0$ is a small number used to transform the strict inequality into inequality constraint $\boldsymbol{\alpha}^{(2)} \geq \boldsymbol{\alpha}^{(1)} + \zeta$.

For complexity and explainability purposes, it is useful to have non-overlapping hyper-rectangle rules. For two hyper-rectangle rules i and j, this is formalised as [Xu et al., 2019]

$$
\begin{aligned}
\max_d \Bigg(\Big| \frac{1}{2} \left(\alpha^{(1)}_{id} + \alpha^{(2)}_{id} \right) - \frac{1}{2} \left(\alpha^{(1)}_{jd} + \alpha^{(2)}_{jd} \right) \Big| \\
- \frac{1}{2} \left(\alpha^{(2)}_{id} - \alpha^{(1)}_{id} \right) - \frac{1}{2} \left(\alpha^{(2)}_{jd} - \alpha^{(1)}_{jd} \right) \Bigg) \geq 0.
\end{aligned}
$$

Enforcing this constraint is a difficult problem in the literature. We tackle it with a simple, yet effective heuristic. Based on the form of the constraint (the maximum is positive when only one of the values is positive), it consists in choosing a specific dimension d and adapting either $\boldsymbol{\alpha}_i$ or $\boldsymbol{\alpha}_j$ along d to satisfy the constraint. The corresponding choices are taken to maximise the expected log-likelihood. Details are discussed in the supplementary material.

3.3 CO-LEARNING STRUGMA AND BLACK-BOX MODELS FOR RULE EXPLANATIONS

This section proposes a co-learning framework where (i) hyper-rectangle rules of STruGMA are learned to globally explain a black-box model and (ii) this black-box is simultaneously constrained by STruGMA to be easier to explain.

3.3.1 Co-learning of the Black-box Model

Let us consider a probabilistic black-box model $p(y|\mathbf{x}; \theta)$ that can be trained with gradient descent. Our goal is to constrain it to follow hyper-rectangle rules of STruGMA as much as possible. This is achieved by using the loss

$$
\lambda \times \mathcal{L}(\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{\theta}) + (1 - \lambda) \times D_{\text{KL}} \big(p(\mathbf{Y}|\mathbf{X}; \boldsymbol{\beta}) || p(\mathbf{Y}|\mathbf{X}; \boldsymbol{\theta}) \big), \tag{3}
$$

where $\lambda \in [0, 1]$ can be a hyper-parameter, $\mathcal{L}(\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{\theta}))$ is a usual loss on training data such as the cross-entropy, D_{KL} is the Kullback–Leibler divergence between the reference model $p(\mathbf{Y}|\mathbf{X}; \boldsymbol{\beta})$ given by STruGMA and the black-box model $p(\mathbf{Y}|\mathbf{X}; \boldsymbol{\theta})$ which is optimised. This divergence acts as a regularisation term that encourages the black-box model to satisfy hyper-rectangle rules of STruGMA. It is a conditional expectation and can be evaluated as

$$
\begin{aligned}
& D_{\text{KL}} \big(p(\mathbf{Y}|\mathbf{X}; \boldsymbol{\beta}) || p(\mathbf{Y}|\mathbf{X}; \boldsymbol{\theta}) \big) = \\
& \qquad \mathbb{E}_{\boldsymbol{x} \sim p(\mathbf{x}; \boldsymbol{\beta})} [D_{\text{KL}}(p(\mathbf{Y}|\boldsymbol{x}; \boldsymbol{\beta}) || p(\mathbf{Y}|\boldsymbol{x}; \boldsymbol{\theta}))] \\
& \qquad \approx \frac{1}{N_s} \sum_{i=1}^{N_s} \sum_{c=1}^{C} p(y = c|\hat{\boldsymbol{x}}_i; \boldsymbol{\beta}) \log \frac{p(y = c|\hat{\boldsymbol{x}}_i; \boldsymbol{\beta})}{p(y = c|\hat{\boldsymbol{x}}_i; \boldsymbol{\theta})},
\end{aligned}
$$

where $\{\hat{\boldsymbol{x}}_i\}_{i=1}^{N_s}$ is a new sample obtained from STruGMA to compute a Monte-Carlo estimate of the divergence term. Sampling from STruGMA has a complexity which is linear with respect to the input space dimension D.

One issue regarding the performance of the learned black-box model, after optimising Eq. 3, is its sensitivity with respect to the choice of λ. Indeed, it can result in a (too) weakly or strongly constrained black-box model. This problem is ubiquitous in multi-objective optimisation. To alleviate this sensible choice of λ, we apply the multiple gradient descent algorithm (MGDA)[Sener and Koltun, 2018, Désidéri, 2009], which consists in finding, at each iteration, the λ^* that gives the direction of gradient that improves both terms of Eq. 3. This λ^* is the one that minimises $||\lambda \nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta}) + (1-\lambda)\nabla_{\boldsymbol{\theta}} g(\boldsymbol{\beta},\boldsymbol{\theta})||$ and is obtained using

$$\lambda^* = \left[\frac{\left(\nabla_{\boldsymbol{\theta}} g(\boldsymbol{\beta},\boldsymbol{\theta}) - \nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta})\right)^\top \nabla_{\boldsymbol{\theta}} g(\boldsymbol{\beta},\boldsymbol{\theta})}{\|\nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta}) - \nabla_{\boldsymbol{\theta}} g(\boldsymbol{\beta},\boldsymbol{\theta})\|_2^2} \right]_{+,\frac{1}{T}}, \quad (4)$$

where $f(\boldsymbol{\theta}) = \mathcal{L}(\boldsymbol{X},\boldsymbol{Y},\boldsymbol{\theta})$, $g(\boldsymbol{\beta},\boldsymbol{\theta}) = \hat{D_{\mathrm{KL}}}\big(p(\mathbf{Y}|\mathbf{X};\boldsymbol{\beta})||p(\mathbf{Y}|\mathbf{X};\boldsymbol{\theta})\big)$, and $[\cdot]_{+,\frac{1}{T}} = \max(\min(.,1),0)$ is a clipping operation to $[0,1]$.

3.3.2 Co-learning of STruGMA through Knowledge Distillation

As we want STruGMA to globally explain the black-box model with hyper-rectangle decision rules, one approach is to use knowledge distillation. Training instances $\boldsymbol{X}$ are relabelled with the outputs $\boldsymbol{Y}_{\boldsymbol{\theta}}$ of the black-box model and STruGMA is learned from this new dataset. As a result, STruGMA approximates the black-box model on the whole input space. The resulting co-learning Algorithm 1 (see also Figure 1) summarises how to learn both models. It has the key advantage to work with any black-box model that is learned through gradient descent.

Algorithm 1 Co-learning of a black-box model with STruGMA

```
Input: training set $\{\mathbf{x}_i, y_i\}_{i=1}^N$, black-box model with parameters $\boldsymbol{\theta}$, number of epochs $N_{\text{epochs}}$, number of gradient descent steps in the M-step $N_{\text{STruGMA}}$, number of rules $K$ per class, size of MC sample $N_s$
Output: trained black-box model and STruGMA
initialise STruGMA with GMMs per class
while not converged do
    // update black-box model (regularised GD)
    draw an MC sample $\{\hat{\boldsymbol{x}}_i\}_{i=1}^{N_s}$ from STruGMA
    get $\lambda*$ from Eq. 4 and train the black-box model with $\lambda^*\mathcal{L}(\boldsymbol{X},\boldsymbol{Y},\boldsymbol{\theta}) + (1-\lambda^*)\hat{D_{\mathrm{KL}}}(p(\mathbf{Y}|\mathbf{X};\boldsymbol{\beta})||p(\mathbf{Y}|\mathbf{X};\boldsymbol{\theta}))$ for $N_{\text{epochs}}$ epochs

    // update STruGMA (knowledge distillation)
    relabel training set with black-box model
    E-step of STruGMA with Eq. 1 (responsabilities)
    M-step of STruGMA with Eq. 2 ($N_{\text{STruGMA}}$ iterations of gradient descent + gradient projection)
end while
return STruGMA and the black box
```

4 EMPIRICAL EVALUATION

Experiments assess whether (i) after co-learning with STruGMA the black-box's decision boundary becomes easier to approximate by a rule learner with a limited impact on its accuracy, (ii) decision rules explanations from distilled decision trees after co-learning are more faithful than those without co-learning and (iii) extracted rules comply with domain knowledge by the means of a qualitative evaluation.

Table 1: Details of the datasets for the experiments.

Dataset	Size	Dimension
Wine	178	13
Pima Indian diabetes (Pima)	768	8
Ionosphere	351	34
Magic gamma (Gamma)	19020	11
Bank marketing (Marketing)	4119	20
German credit (Credit)	1000	20
Waveform	5000	40

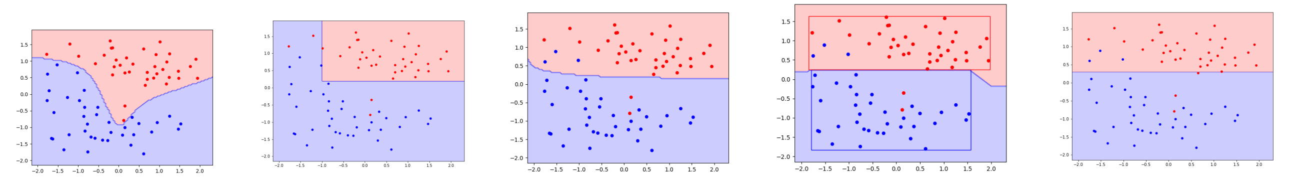
4.1 EXPERIMENTAL SETTINGS

We validate our method[2] on a synthetic dataset and on seven commonly used machine learning datasets from UCI [Dua and Graff, 2017] for which neural networks usually outperform decision trees; see Table 1 for their details.

We chose deep neural networks with architectures inspired from the literature [Wu et al., 2020, Ribeiro et al., 2018, Pedapati et al., 2020], as they work well for tabular data. Our dense hidden layers have *ELUs* as activation functions and the last layer has the dimension of the number of classes. Details about these architectures are left in the supplementary material. At each iteration of the co-learning, only $N_{\text{epochs}} = 1$ epochs are spent for updating the MLP. Indeed, if the complexity of the black-box model changes too rapidly, STruGMA will not be able to explain it and the D_{KL} term will not be able to play its role as a complexity regulariser. The number of iterations $N_{\text{STruGMA}} = 100$ for the M-step ensures that STruGMA is a good approximation of the black-box model at each iteration. The size of the Monte Carlo sample to approximate the divergence between the two models is set to $N_s = 10 \times N$. The margin ζ between $\boldsymbol{\alpha}^{(1)}$ and $\boldsymbol{\alpha}^{(2)}$ is set to a relatively small value of 0.2. A Gaussian mixture model is used to initialise STruGMA, with $\boldsymbol{\alpha}_k^{(1)} = \boldsymbol{\mu}_k - 0.2\boldsymbol{\sigma}_k$ and $\boldsymbol{\alpha}_k^{(2)} = \boldsymbol{\mu}_k + 0.4\boldsymbol{\sigma}_k$ for each component. The number of components K is the same for each STruGM per class and is chosen in $\{2,3,4\}$ to avoid complex rule explanations. This hyper-parameter is chosen with a separate validation set. Both STruGMA and the black-box model are learned using the gradient-based optimiser Adam [Kingma and Ba, 2015] with 10^{-3} as learning rate.

We use the accuracy (percentage of correct predictions) to assess the quality of predictions and the fidelity (percentage of predictions where a black-box and a white-box model

[2] Our implementation is available at `https://github.com/gerald4/Co-learning_with_STruGMA`.



(a) Two-layer MLP without co-learning (b) TreeExplainer for the MLP in Figure 3a (c) Two-layer MLP co-learned with STruGMA (d) STruGMA co-learned with the MLP in Figure 3c (e) TreeCoExplainerHR with STruGMA splits

Figure 3: Decision boundary of several models, (a-b) without and (c-e) with co-learning, on a two-dimensional example. The black box model is easier to explain with rules after co-learning (c) than in the standard case (a).

agree) to measure the mutual agreement.

It is possible to directly use hyper-rectangle rules of STruGMA as explanations when the input space dimension D is relatively low. However, to get simple rules with limited size, we resort to decision trees as rule learners to provide global explanations. When co-learning is done, similarly as post-hoc methods, a *distilled* decision tree is trained by mimicking predictions of the co-learned black-box model. We explore two possibilities: either we use the predictions of the co-learned black-box model with original features to train a distilled decision tree (TreeCoExplainerBB) or we use the predictions of the co-learned black-box model with hyper-rectangle splits transformed as binary features to train the distilled decision tree (TreeCoExplainerHR). Both are compared with a baseline (TreeExplainer) where the distilled decision tree is trained to mimic the black-box model without co-learning. This baseline is representational of the global post-hoc (through decision trees) explanation methods discussed in Section 2.1.

4.2 EFFECT OF CO-LEARNING ON MODEL INFERENCE

Figure 3 illustrates the effect of co-learning on a synthetic two-dimensional toy example. Figure 3a shows a black-box model learned without any kind of co-learning, returning a decision boundary which is somewhat complex for a simple toy problem. A distilled decision tree is then learned in Figure 3b to explain this black-box model. Although the region where $x_1 < -1$ and $x_2 > 1.1$ contains training instances, the decision tree fails to explain it correctly. This problem is avoided with the co-learning of the black-box model in Figure 3c and STruGMA in Figure 3d. The corresponding distilled decision tree (our TreeCoExplainerHR) in Figure 3e globally explains the co-learned black-box model. This toy example illustrates how co-learning rectifies the decision boundary of a black-box model to be compatible with rules. The co-learned black-box model is very likely to follow rule explanations extracted by rule learners such as decision trees. Note that the black-box model in Figure 3a may be more accurate than its co-learned version in Figure 3c.

4.3 IMPACT ON FIDELITY AND ACCURACY

Table 2 shows the test fidelity of the baseline TreeExplainer and our methods TreeCoExplainerHR and TreeCoExplainerBB. Tree depth is chosen on a separate validation set. On all datasets, results show that co-learning improves fidelity between the black-box model and distilled decision trees. This means that one can be more confident in explanations based on decision trees after co-learning with STruGMA than without co-learning.

Table 3 shows the accuracy of the black-box model (BB) without co-learning and the co-learned black-box model (coBB). It can be seen that the co-learning usually negatively impacts the test accuracy of the black-box model (except on Credit and Ionosphere). However, the difference is usually not important, as it is usually around 2%. This difference is also perceived on the accuracy of distilled decision trees in Table 4. Nonetheless, as it can be seen in Table 5, our co-learned black-box models still usually perform better

Table 2: Impact of co-learning on the test fidelity. Mean and standard deviation over 10 repetitions are reported. TreeExplainer is the distilled decision tree obtained with the predictions of the black-box model without co-learning, whereas TreeCoExplainerHR (resp. TreeCoExplainerBB) is our distilled decision tree of the co-learned black-box model using the hyper-rectangle rules of STruGMA as binary features (resp. original features).

Dataset	TreeExplainer	TreeCoExplainerHR	TreeCoExplainerBB
Bank	95.97 (0.74)	96.18 (0.63)	**96.49 (0.89)**
Credit	77.3 (3.47)	81.25 (3.47)	**81.5 (3.43)**
Ionosphere	87.32 (3.25)	**90.28 (3.42)**	88.87 (5.69)
Gamma	93.31 (2.08)	93.15 (0.85)	**95.6 (0.36)**
Pima	88.44 (2.41)	88.9 (1.35)	**92.01 (3.24)**
Waveform	80.26 (1.53)	80.52 (1.87)	**80.86 (1.28)**
Wine	89.17 (4.62)	**92.78 (4.93)**	89.72 (2.64)

than an interpretable decision tree. Overall, in addition to providing faithful global explanations thanks to co-learning with STruGMA, our coBB models remain competitive in terms of predictive performance compared to decision trees.

Table 3: Predictive accuracy of co-learned black-box models (coBB) and black-box models without co-learning (BB). Mean and standard deviation are shown over 10 repetitions.

Dataset	coBB	BB
Bank	90.68 (0.77)	**90.99 (0.84)**
Credit	**75.65 (3.88)**	74.75 (3.5)
Ionosphere	**90.98 (3.88)**	90.56 (3.45)
Gamma	80.57 (0.49)	**82.79 (2.53)**
Pima	73.12 (2.31)	**75.39 (1.77)**
Waveform	85.97 (0.87)	**86.15 (0.7)**
Wine	96.94 (2.43)	**97.5 (2.05)**

Table 4: Predictive accuracy of distilled trees. Mean and standard deviation over 10 repetitions are reported.

Dataset	TreeExplainer	TreeCoExplainerHR	TreeCoExplainerBB
Bank	**91.29 (0.94)**	90.42 (1.0)	90.81 (0.99)
Credit	69.15 (3.33)	71.5 (2.59)	**71.55 (4.7)**
Ionosphere	**88.03 (3.89)**	87.18 (4.01)	86.34 (3.45)
Gamma	**80.79 (2.09)**	77.04 (1.12)	79.16 (0.37)
Pima	**72.4 (1.44)**	71.24 (3.17)	71.88 (2.12)
Waveform	76.43 (1.9)	76.38 (1.83)	**76.71 (1.58)**
Wine	89.17 (3.33)	**91.67 (4.54)**	88.89 (3.93)

Table 5: Predictive accuracy of co-learned black-box models (coBB) and decision trees (DT). Mean and standard deviation over 10 repetitions are reported.

Dataset	coBB	DT
Bank	90.68 (0.77)	**90.81 (0.96)**
Credit	**75.65 (3.88)**	71.05 (3.3)
Ionosphere	**90.98 (3.88)**	90.28 (4.43)
Gamma	80.57 (0.49)	**82.72 (0.43)**
Pima	**73.12 (2.31)**	72.02 (2.59)
Waveform	**85.97 (0.87)**	75.24 (1.23)
Wine	**96.94 (2.43)**	87.78 (4.93)

4.4 EVOLUTION OF THE DISTANCE BETWEEN THE BLACK-BOX MODEL AND STRUGMA

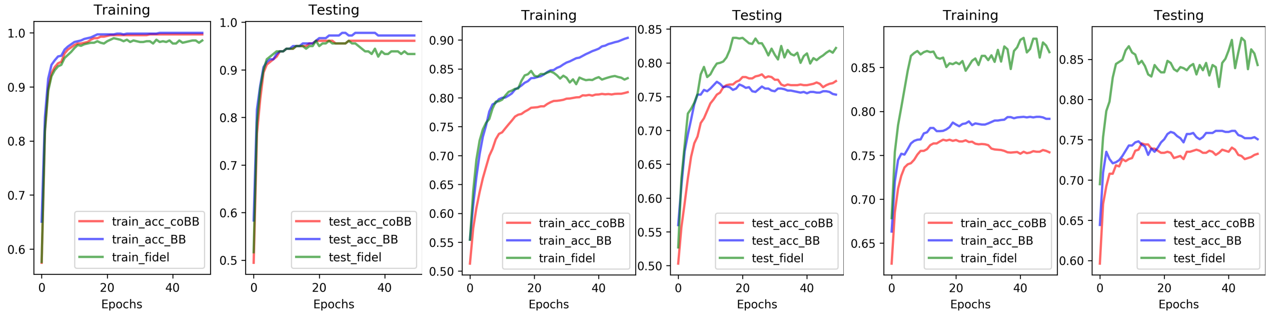
Figure 4 shows the evolution of the accuracy and fidelity over the first 50 co-learning iterations for each dataset. Despite the iterative nature of the co-learning that alternates between learning the black-box model and its STruGMA surrogate, the fidelity of the two models increases throughout iterations. This means that the main goal which is essentially to minimise the distance between the two models can be achieved with co-learning. Moreover, in Figure 5, the losses decrease properly and a local optimal can usually be reached after or even before 50 iterations of co-learning.

4.5 RELEVANCE OF DECISION RULES EXPLANATIONS

We finish our experiments with a study of 2 use cases in the medical domain where a black-box model is learned through co-learning and TreeCoExplainerHR is used to globally explain it. The first use case is to diagnose patients in the *heart disease* (Cleveland) dataset [Dua and Graff, 2017]. The second use case studies the survival after one year of patients who underwent major lung resections for primary lung cancer in the *thoracic surgery* dataset. Indeed, according to Doshi-Velez and Kim [2017], one way to measure interpretability is to get feedback from a relevant domain expert. We, therefore, asked a medical doctor if these explanations are convincing and can be clinically accepted.

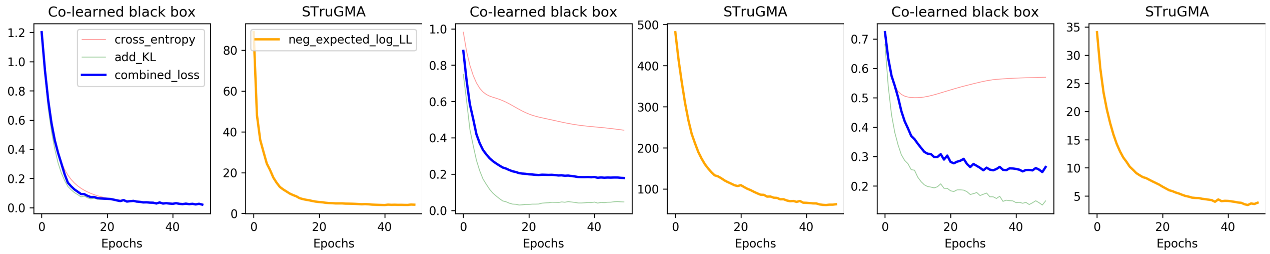
Table 6 (a) shows the explainer TreeCoExplainerHR of a co-learned black-box model. The first remark on rules (first, second and third) in the left part is that although the patient is asymptomatic (no chest pain), the thallium stress test reveals malformations on the heart vessels. Therefore, if it is reversible (i.e., it can be fixed, but has not already been fixed) or it is fixed but there still exist damaged heart vessels, the patient has heart disease. According to the medical doctor, these rules are completely in accordance with medical knowledge because the chosen attributes and their values are clinically correct. Nonetheless, the fourth rule was not very satisfactory because the medical doctor would have expected other attributes. Furthermore, according to the medical doctor, the fifth rule is the typical patient that cardiologists usually encounter every day because of damaged major vessels damaged and thallium test. The sixth rule was also clinically correct because the patient is symptomatic (i.e., suffers); if he has damaged heart vessels, it is obvious (according to the doctor) that he has the disease.

For the second use case in Table 6 (b), according to the medical doctor, the first and third rules are fully compliant with clinical requirements, as they can be considered as *rules of thumb*. Indeed, if a patient has a large tumour (diameter greater than 1cm), surgery should be avoided. The same applies to patients who have tumours with metastases



(a) Wine (b) Credit (c) Pima

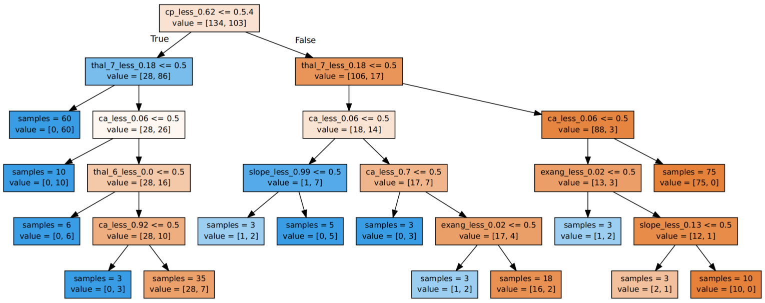
Figure 4: Evolution of mean of accuracy and fidelity over five repetitions of the first 50 iterations of co-learning.



(a) Wine (b) Credit (c) Pima

Figure 5: Evolution of the losses over the first 50 iterations of co-learning (one repetition/dataset). Cross_entropy is the cross-entropy of the co-learned black-box model whereas add_KL is the divergence between the same model and STruGMA. Combined_loss is the convex combination of cross_entropy loss and the add_KL loss whereas neg_expected_log_LL is the negative expected log-likelihood of the STruGMA. Plots (a–c) share the same legend.

Table 6: Distilled decision trees TreeCoExplainerHR and decision rules for heart disease (top) and thoracy surgery (bottom).



IF Chest.pain.type = asymptomatic **AND** Thalium.test = reversible **THEN disease (60/60)**

IF Chest.pain.type = asymptomatic **AND** Thalium.test = normal or fixed.defect **AND** Major.vessel.damaged = 1 **THEN disease (10/10)**

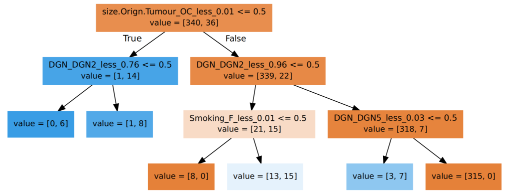
IF Chest.pain.type = asymptomatic **AND** Thalium.test = fixed.defect **AND** Major.vessel.damaged = 0 or 2 or 3 **THEN disease (6/6)**

IF Chest.pain.type = asymptomatic **AND** Thalium.test = normal **AND** Major.vessel.damaged = 3 **THEN disease (6/6)**

IF Chest.pain.type = typical.angina or atypical.angina or non-anginal.pain **AND** Thalium.test = reversible **AND** Major.vessel.damaged = 1 or 3 **THEN disease (10/11)**

IF Chest.pain.type = typical.angina or atypical.angina or non-anginal.pain **AND** Thalium.test = reversible **AND** Major.vessel.damaged = 0 or 2 **AND** Exercise.induced.angina = 1 **THEN disease (2/3)**

OTHERWISE no disease



IF Size.origin.tumor = largest **THEN die (14/15)**

IF Size.origin.tumor = smallest or small or large **AND** Tumor.diagnosic = secondary **AND** Smoking = True **THEN die (15/28)**

IF Size.origin.tumor = smallest or small or large **AND** Tumor.diagnosic = multiple **THEN die (7/10)**

OTHERWISE survive

(multiple tumours). He was able to understand the second rule, but he did not fully accept it as he expected to see other attributes to decide to accept or reject the explanation.

In summary, for the doctor that we interviewed, even though he does not know how the complex black-box model works, thanks to TreeCoExplainerHR, he was able to understand the logic behind predictions and, more importantly, to make connections with similar patients he already consulted.

5 CONCLUSION

This paper proposes a co-learning framework for global explanations of black-box models with decision rules. In this framework, a black-box model is explained by co-learning a newly introduced soft truncated Gaussian mixture analysis (STruGMA) that encapsulates hyper-rectangle decision rules. Simultaneously, the black-box model is encouraged by a penalty term to satisfy the hyper-rectangle rules of STruGMA. Results show that our framework improves the fidelity of global explanations, while having a limited impact on the accuracy of the black-box model, which remains competitive. Yet, in further works, the user interview in Section 4.5 should be extended as a user study to compare the interest of rules learned with or without co-learning. Our framework also opens up a wide range of perspectives since it can be used for any black-box model trainable through gradient descent. Future works will consider other black-box models like SVMs and other rule learners to provide global explanations. In addition, one can directly inject strong priors on STruGMA to automatically get decision rules explanations. Finally, it will be interesting to perform experiments with image data to provide more faithful global explanations of deep CNNs in the light of Zhang et al. [2019].

Acknowledgements

This work has been funded by the EOS-VeriLearn, project number 30992574 of the Fonds de la Recherche Scientifique (F.R.S-FNRS) in Belgium. The authors thank Adrien Bibal, Jérôme Fink and Minh Viet Vu for their fruitful comments and suggestions that helped to finalise the paper.

References

A Duygu Arbatli and H Levent Akin. Rule extraction from trained neural networks using genetic algorithms. *Nonlinear Analysis: Theory, Methods & Applications*, 30(3): 1639–1648, 1997.

Stephen Boyd, Stephen P Boyd, and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

Olcay Boz. Extracting decision trees from trained neural networks. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 456–461, New York, NY, USA, 2002. ACM Press.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020.

A Clifford Cohen Jr. Estimating the mean and variance of normal populations from singly truncated and doubly truncated samples. *The Annals of Mathematical Statistics*, pages 557–569, 1950.

R. Confalonieri, T. Weyde, T. R. Besold, and F. Moscoso del Prado Martín. Trepan reloaded: A knowledge-driven approach to explaining artificial neural networks. In *24th European Conference on Artificial Intelligence*, volume 325 of *Frontiers in Artificial Intelligence and Applications*, pages 2457–2464. IOS Press, 2020.

Mark Craven and Jude W Shavlik. Using sampling and queries to extract rules from trained neural networks. In *Proceedings of the Eleventh International Conference on International Conference on Machine Learning*, pages 37–45. 1994.

Mark W. Craven and Jude W. Shavlik. Extracting tree-structured representations of trained networks. In *Proceedings of the 8th International Conference on Neural Information Processing Systems*, pages 24–30, Cambridge, MA, USA, 1995. MIT Press.

Mark William Craven and Jude W. Shavlik. *Extracting Comprehensible Models from Trained Neural Networks*. PhD thesis, 1996. AAI9700774.

Jean-Antoine Désidéri. *Multiple-Gradient Descent Algorithm (MGDA)*. PhD thesis, INRIA, 2009.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, 2019.

Finale Doshi-Velez and Been Kim. Towards a rigorous science of interpretable machine learning. *arXiv*, 2017. URL https://arxiv.org/abs/1702.08608.

Dheeru Dua and Casey Graff. UCI machine learning repository, 2017. URL http://archive.ics.uci.edu/ml.

Johannes Fürnkranz, Dragan Gamberger, and Nada Lavrač. *Foundations of rule learning*. Springer Science & Business Media, 2012.

Divya Gopinath, Hayes Converse, Corina Pasareanu, and Ankur Taly. Property inference for deep neural networks. In *2019 34th IEEE/ACul International Conference on Automated Software Engineering (ASE)*, 2019.

Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. A survey of methods for explaining black box models. *ACM Comput. Surv.*, 51(5), August 2018.

Trevor Hastie and Robert Tibshirani. Discriminant analysis by gaussian mixtures. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):155–176, 1996.

Zhiting Hu, Zichao Yang, Ruslan Salakhutdinov, and Eric Xing. Deep neural networks with massive learned knowledge. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1670–1679, 2016.

Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, et al. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *International Conference on Machine Learning*, pages 2668–2677, 2018.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations*, San Diego, CA, USA, 2015.

Günter Klambauer, Thomas Unterthiner, Andreas Mayr, and Sepp Hochreiter. Self-normalizing neural networks. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, page 972–981, Red Hook, NY, USA, 2017.

Himabindu Lakkaraju, Stephen H. Bach, and Jure Leskovec. Interpretable decision sets: A joint framework for description and prediction. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, page 1675–1684, New York, NY, USA, 2016. Association for Computing Machinery.

Scott M Lundberg, Gabriel Erion, Hugh Chen, Alex DeGrave, Jordan M Prutkin, Bala Nair, Ronit Katz, Jonathan Himmelfarb, Nisha Bansal, and Su-In Lee. From local explanations to global understanding with explainable ai for trees. *Nature machine intelligence*, 2(1):2522–5839, 2020.

David Alvarez Melis and Tommi Jaakkola. Towards robust interpretability with self-explaining neural networks. In *Advances in Neural Information Processing Systems*, pages 7775–7784, 2018.

Graziano Mita, Paolo Papotti, Maurizio Filippone, and Pietro Michiardi. LIBRE: Learning interpretable boolean rule ensembles. In *23rd International Conference on Artificial Intelligence and Statistics, 3-5 June 2020, Palermo, Sicily, Italy*, Palermo, ITALY, 06 2020.

Yuzuru Okajima and Kunihiko Sadamasa. Deep neural networks constrained by decision rules. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 2496–2505, 2019.

Tejaswini Pedapati, Avinash Balakrishnan, Karthikeyan Shanmugam, and Amit Dhurandhar. Learning global transparent models consistent with local contrastive explanations. In *Advances in Neural Information Processing Systems*, pages 3592–3602, 2020.

Dino Pedreschi, Fosca Giannotti, Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, and Franco Turini. Meaningful explanations of black box ai decision systems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 9780–9784, 2019.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Model-agnostic interpretability of machine learning. In *ICML Workshop on Human Interpretability in Machine Learning*, Stockholm, Sweden, 2016.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Anchors: High-precision model-agnostic explanations. In *AAAI Conference on Artificial Intelligence*, 2018.

Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 525–536, 2018.

Dylan Slack, Sophie Hilgard, Emily Jia, Sameer Singh, and Himabindu Lakkaraju. Fooling lime and shap: Adversarial attacks on post hoc explanation methods. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, AIES '20, page 180–186, New York, NY, USA, 2020.

Allyson Souris, Anirban Bhattacharya, and Debdeep Pati. The soft multivariate truncated normal distribution. *arXiv preprint arXiv:1807.09155*, 2018.

Sicco Verwer and Yingqian Zhang. Learning optimal classification trees using a binary linear program formulation. In *33rd AAAI Conference on Artificial Intelligence*, 2019.

Lior Wolf, Tomer Galanti, and Tamir Hazan. A formal approach to explainability. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, page 255–261, New York, NY, USA, 2019. Association for Computing Machinery.

Mike Wu, Michael C Hughes, Sonali Parbhoo, Maurizio Zazzi, Volker Roth, and Finale Doshi-Velez. Beyond sparsity: Tree regularization of deep models for interpretability. In *AAAI Conference on Artificial Intelligence*, 2018.

Mike Wu, Sonali Parbhoo, Michael C. Hughes, Ryan Kindle, Leo A. Celi, Maurizio Zazzi, Volker Roth, and Finale Doshi-Velez. Regional tree regularization for interpretability in deep neural networks. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence*, pages 6413–6421, February 7-12 2020.

Taufik Xu, LI Chongxuan, Jun Zhu, and Bo Zhang. Multi-objects generation with amortized structural regularization. In *Advances in Neural Information Processing Systems*, pages 6619–6629, 2019.

Hongyu Yang, Cynthia Rudin, and Margo Seltzer. Scalable bayesian rule lists. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, page 3921–3930, 2017.

Quanshi Zhang, Yu Yang, Haotian Ma, and Ying Nian Wu. Interpreting cnns via decision trees. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6261–6270, 2019.

Ying Zhang, Tao Xiang, Timothy M Hospedales, and Huchuan Lu. Deep mutual learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4320–4328, 2018.