Datenbank Spektrum (2020) 20:171–180
https://doi.org/10.1007/s13222-020-00351-x

SCHWERPUNKTBEITRAG

Towards Understanding and Arguing with Classifiers: Recent Progress

Xiaoting Shao[1] · Tjitze Rienstra[1] · Matthias Thimm[1] · Kristian Kersting[1]

Received: 19 February 2020 / Accepted: 11 June 2020 / Published online: 2 July 2020

Abstract
Machine learning and argumentation can potentially greatly benefit from each other. Combining deep classifiers with knowledge expressed in the form of rules and constraints allows one to leverage different forms of abstractions within argumentation mining. Argumentation for machine learning can yield argumentation-based learning methods where the machine and the user argue about the learned model with the common goal of providing results of maximum utility to the user. Unfortunately, both directions are currently rather challenging. For instance, combining deep neural models with logic typically only yields deterministic results, while combining probabilistic models with logic often results in intractable inference. Therefore, we review a novel deep but tractable model for conditional probability distributions that can harness the expressive power of universal function approximators such as neural networks while still maintaining a wide range of tractable inference routines. While this new model has shown appealing performance in classification tasks, humans cannot easily understand the reasons for its decision. Therefore, we also review our recent efforts on how to "argue" with deep models. On synthetic and real data we illustrate how "arguing" with a deep model about its explanations can actually help to revise the model, if it is right for the wrong reasons.

Keywords Argumentation-based ML · Explainable AI · Interactive ML · Influence Function · Deep Density Estimation · Probabilistic Circuits

1 Introduction

Classification is the problem of categorizing new observations by using a classifier learnt from already categorized examples. In general, the area of machine learning has brought forth a series of different approaches to deal with this problem, from decision trees over support vector machines to deep neural networks. Recently, approaches to statistical relational learning [6] even take the perspective of knowledge representation and reasoning into account by developing models on more formal logical and statistical grounds. One can even combine the latter with deep learning into a single system. The resulting *neural-symbolic* systems such as DeepProbLog [20] are capable of modeling knowledge and constraints with a logic formalism, while maintaining the computational power of deep neural. One can even integrate probabilistic circuits such as sum-product network [35], featuring deep hierarchical models with tractable inference.

These developments impact both computational models of argumentation [3] and argumentation mining [19]. In computational argumentation, structured arguments have been studied and formalized for decades using models that can be expressed in a logic framework. At the same time, argumentation mining has rapidly evolved by exploiting state-of-the-art neural architectures coming from deep learning. However, these two worlds have progressed largely independently of each other. Only recently, a few works have taken some steps towards the integration of such methods, by applying techniques combining sub-symbolic classifiers with knowledge expressed in the form of rules and constraints to argumentation mining, see e.g. [10]. Moreover,

We only sketch and review our recent efforts. More details can be found in the corresponding publications [9, 31, 33, 37] and current submissions to conferences and journals.

✉ Xiaoting Shao
xiaoting.shao@cs.tu-darmstadt.de

Tjitze Rienstra
rienstra@uni-koblenz.de

Matthias Thimm
thimm@uni-koblenz.de

Kristian Kersting
kersting@cs.tu-darmstadt.de

[1] Technische Universität Darmstadt, Darmstadt, Germany

argumentation-based machine learning employs computational models of argumentation for reasoning within machine learning itself [23, 28, 39]. For instance, Thimm and Kersting [39] proposed a two-step classification approach. In the first step, rule learning algorithms are used to extract frequent patterns and rules from a given data set. The output of this step comprises a huge number of rules (given fairly low confidence and support parameters) and these cannot directly be used for the purpose of classification as they are usually inconsistent with one another. Therefore, in the second step, they interpret these rules as the input for approaches to structured argumentation. This allows one to obtain classifiers, which are by design able to explain their decisions, and therefore address the recent need for Explainable AI: classifications are accompanied by a dialectical analysis showing why arguments for the conclusion are preferred to counterarguments. Argumentation techniques in machine learning also allows the easy integration of additional expert knowledge in form of arguments.

While these results on combining machine learning and argumentation are encouraging, there are still many challenges. Consider e.g. neural-symbolic systems. While deep neural networks are highly expressive, they typically yield only deterministic results. In contrast, (deep) density estimators can model uncertainty, but (marginal) inference is in general intractable. Indeed, probabilistic circuits such as sum-product networks (SPNs) [26] provide tractable inference, but unfortunately, they are generally not universal function approximators [4]. Therefore, we recently proposed conditional sum-product networks (CSPNs) [33] that can harness the expressive power of universal function approximators such as neural networks, while still maintaining a wide range of probabilistic inference routines. Empirically, CSPNs achieve appealing performance in classification tasks.

Moreover, the high predictive performance of highly expressive deep classifiers raises the question whether we can actually trust them by only looking at the accuracy. Just because a machine learning model is highly accurate does not mean it represents the right mapping. Consider the recent study due to Lapuschkin *et al.* on what machine learning models really learn [16]. This study observed that a deep neural network trained on the PASCAL VOC 2007 data set [8] focuses actually on source tags, which incidentally correlate with the labels, for prediction. This “Clever Hans”-like moments [32] happens when the model has learnt spurious artifacts, also known as confounding factors. Especially in real-world domains that are typically high dimensional, collecting ”enough“ data is often very expensive or even impossible. In this case the data is prone to spurious artifacts, which could be accidentally learnt by the models [2]. When the model’s underlying behavior is systematically wrong, it may not generalize well to unseen data. Systematic wrong behavior can be hard to spot and do real harm. For instance, Obermeyer *et al.* [25] revealed that a widely-used commercial model for predicting medical needs exhibits significant racial bias where black patients are considerably sicker than white patients, at a given risk score. This is attributed to the fact that the model uses medical expenses to predict medical needs, however, black people have less access to medical care, which means fewer medical expenses are given to them compared to white people. This racial bias in the model could pose a real danger to black patients. While using Explainable AI or making even deep learning explainable by design, for instance using argumentation-based machine learning, may help to discover the bias, the true goal is to eliminate bias. To this end, we add the expert into the training loop such that she starts to argue with the model by providing feedback on its arguments for classification, i.e., explanations.

In the following we will briefly inform about our work conducted towards understanding and “arguing” with classifiers within the ”Argumentative Machine Learning” (CAML) project as part of the SPP “RATIO”. Generally, CAML aims for a general argumentation framework. Towards this end, we extend e.g. rule mining algorithms to extract rules from statistical models, and we consider interactive explanations in machine learning as a new form of argumentation. We proceed as follows. First, we review the definition and learning algorithm for conditional sum-product networks in Sect. 2 along with some empirical evaluations. Then we review our work on interactively correcting differentiable classification models in Sect. 3, and we show the effectiveness of our method empirically.

2 A novel tractable deep probabilistic classifier

Argumentation Mining aims at identifying and interpreting argument components out of input text [19]. For example, if we take a basic claim-premise argument model, possible tasks could be claim detection [1, 18], evidence detection [27], and the prediction of links between claim and evidence [11, 24]. One way to exploit domain knowledge in argumentation mining is to apply a set of hand-engineered rules on the output of some first stage classifier (such as a neural network). NeSy or SRL approaches can impose those rules as constraints during training to ensure that solutions are consistent with those rules. Therefore, if a neural network is trained to classify argument components, and another one is trained to detect links between them, additional global constraints can be enforced to adjust the weights of the networks toward admissible solutions. We refer to [10] for implementation examples with DeepProbLog and with GS-MLNs. Sum-Product Logic [35] even features deep hi-

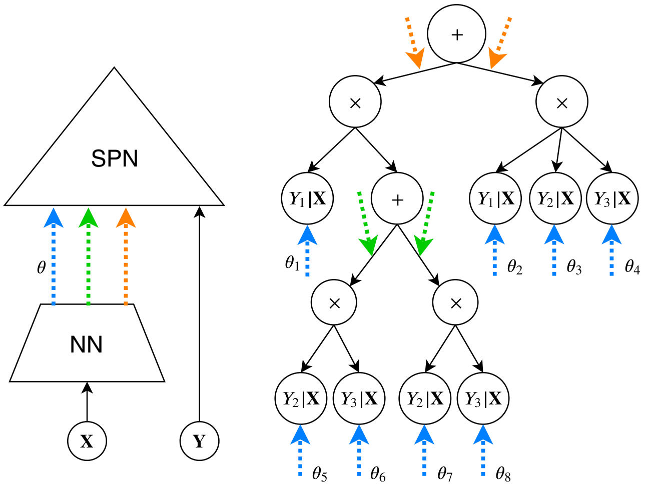

Fig. 1 Overview of the architecture (*left*) and a concrete CSPN example encoding $P(\mathbf{Y}\,|\,\mathbf{X})$ (right). $\mathbf{X}$ is the set of conditional variables and $\mathbf{Y}$ consists of three RVs Y_1, Y_2 and Y_3. Each color of the arrow represents one data flow. Here, the gating weights, possibly also leaf nodes, are parameterized by the output of neural networks given $\mathbf{X}$. Taken from [33]

erarchical models with tractable inference within neural-symbolic AI.

However, as argued above, we may want to put some (conditional) structure into neural-symbolic approaches, which may also be improved iteratively as we show later.

To this end, we develop conditional sum-product networks (CSPNs), which is a conditional variant of sum-product networks (SPNs). We formally defined CSPNs, provided a learning framework for them, and provided arguments for why CSPNs are more compact than SPNs.

Definition of Conditional SPNs (CSPNs). Specifically, a CSPN as a rooted DAG containing three types of nodes, namely *leaf*, *gating*, and *product* nodes, encoding a conditional probability distribution $P(\mathbf{Y}\,|\,\mathbf{X})$. See Fig. 1 for an illustrative example of a CSPN. Each leaf encodes a normalized univariate conditional distribution $P(Y\,|\,\mathbf{X})$ over a target random variable (RV) $Y \in \mathbf{Y}$, where Y is denoted as the leaf's *conditional scope*. One can also realize neural CSPNs, which rely on random SPN structures parameterized by the output of deep neural networks. While this approach does not have the benefit of carefully learned structures, it gains expressiveness through increased model size. See Fig. 1 for this architecture illustration.

(Structure) Learning CSPNs. To learn CSPNs, we proposed a *LearnCSPN* routine that builds a CSPN top-down by introducing nodes while partitioning a data matrix whose rows represent samples and columns RVs in a recursive and greedy manner. LearnCSPN creates one of the three node types at each step: (1) a leaf, (2) a product, or (3) a gating node. If only one target RV Y is present, one conditional probability distribution can be fit as a leaf. To generate product nodes, conditional independencies are found by means of a statistical test to partition the set of target RVs $\mathbf{Y}$. If no such partitioning is found, then training samples are partitioned into clusters (conditioning) to induce a gating node.

Specifically, we use Generalized Linear Models (GLMs) [21] in the leaves to model univariate distribution but note that *any* univariate tractable conditional model can be plugged into a CSPN effortlessly in order to model $P(Y\,|\,\mathbf{X})$. That is, we compute $P(y\,|\,\mu = \mathrm{glm}(\mathbf{X}))$ by regressing univariate parameters μ from features $\mathbf{X}$, for a given set of distributions in the exponential family.

For product nodes, we are interested in decomposing the labels $\mathbf{Y}$ into subsets that are independent given $\mathbf{X}$. Since we aim to accommodate arbitrary leaf conditional distributions in CSPNs, regardless of their parametric likelihood models or data types (i.e. discrete or continuous), we adopt a non-parametric pairwise conditional independence (CI) test procedure to decompose labels $\mathbf{Y}$. Specifically, we employ randomized conditional correlation test (RCoT). We refer to [36] for further details. After we get the pairwise conditional independence on $\mathbf{Y}$, we create a graph where the nodes are RVs in $\mathbf{Y}$ and put an edge between two nodes Y_i, Y_j if we cannot reject the null hypothesis that $Y_i \perp\!\!\!\perp Y_j\,|\,\mathbf{X}$ for a given threshold α. The conditional scopes of product children are then given by connected components of this graph, akin to [12].

Finally, gating nodes represent a mixture of $\mathbf{Y}$ conditioned on $\mathbf{X}$ weighted by a gating function $g_k(\mathbf{X})$. Ideally, we select a differentiable parametric function, such as logistic regression or a neural network, as the gating function. This function is restricted to allow for a proper mixture of distributions, i.e., $\sum_k g_k(\mathbf{X}) = 1$ and $\forall_{\mathbf{X}} g_k(\mathbf{X}) \geq 0$. To learn the components of the mixture, we perform clustering over features $\mathbf{X}$, and denote the corresponding member assignment as a one-hot coded vector $\mathbf{Z}$. We then proceed to fit the gating function to predict $\mathbf{Z}_k = g_k(\mathbf{X})$.

Having a structure, one can estimate the parameters of the CSPNs, i.e., the weights for the gating nodes and the distributional parameters for the leaf nodes. During structure learning, we learn the parameters automatically with the structure. However, those parameters are only locally optimized and usually not optimal for the global distribution. Since CSPNs are differentiable, we can maximize the overall conditional likelihood in an end-to-end fashion using gradient-based optimization techniques after structure learning. An alternative for learning CSPNs is to start with a random structure, and initialize all the parameters randomly as well, then directly conduct parameter optimization end-to-end.

Autoregressive SPN. CSPNs can be naturally combined with other CSPNs and SPNs to impose a rich structure on high-dimensional joint distributions. We illustrate this by

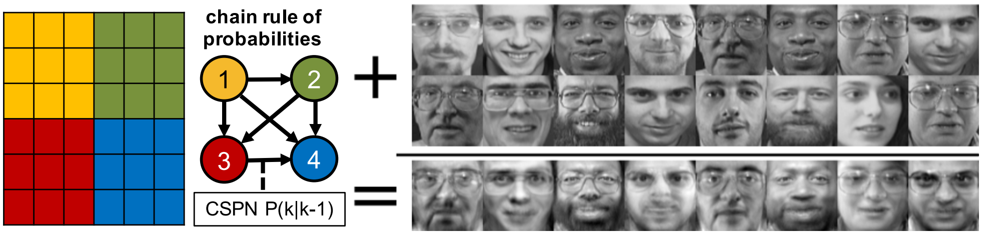

Fig. 2 Imposing structure on deep probabilistic architectures. (*Left*) An Autoregressive Block-wise CSPN (ABCSPN) factorizes a distribution over images along image patches. (*Right*) Conditional image generation with ABCSPNs: bottom row images are sampled while conditioning on the two classes to which individuals from the two upper rows belong. Taken from [33]

introducing ABCSPNs, i.e. autoregressive SPNs for conditional image generation. That is, we model images block by block and decompose the joint image distribution into a product of (C)SPNs, cf. Fig. 2 (left). We investigated ABCSPNs on a subset (20000 random samples) of MNIST and Olivetti faces by splitting each image into 16 resp. 64 blocks of equal size where we normalized the greyscale value for MNIST. Then we trained a CSPN on Gaussian domain for each block conditioned on all the blocks above and to the left of it and on the image class and formulate the distribution of the images as the product of all the CSPNs. As can be seen in Fig. 2 (right), samples from ABCSPNs look quite plausible.

Multi-Label Classification. To further demonstrate the efficiency of CSPNs, we consider multi-label classification. This is a generalization of the classical multi-class classification, which is the single-label problem of categorizing instances into precisely one of more than two classes. In multi-label classification there is no constraint on how many of the classes the instance can be assigned to. We evaluated CSPNs on several multilabel image classification tasks. The goal of each task was to predict the joint conditional distribution of binary labels Y given an image X. Experiments were conducted on the CelebA data set, which features images of faces annotated with 40 binary attributes. In addition, we constructed multilabel versions of the MNIST and Fashion-MNIST data sets, by adding additional labels indicating symmetry, size, etc. to the existing class labels, yielding 16 binary labels total.

We compared CSPNs to two different common ways of parameterizing conditional distributions using neural networks. The first is the mean field approximation. Second, we compared to mixture density networks with 10 mixture components, each itself a mean field distribution. The resulting conditional log-likelihoods as well as accuracies are given in Tab. 1. The results indicate that the commonly used mean field approximation is inappropriate on the considered data sets, as allowing the inclusion of conditional dependencies resulted in a pronounced increase in both likelihood and accuracy. In addition, the improved model capacity of the CSPN compared to the MDN yielded a further performance increase. On CelebA, our CSPN outperforms a number of sophisticated neural network architectures from the literature, despite being based on a standard convnet with only about 400k parameters [7].

Poisson Distributions. Finally, CSPNs are not restricted to binary or Gaussian output distributions. They can also encode multi-variate conditional distributions of other statistical types. We considered temporal vehicular traffic flows from [14], where the data represents the count of vehicles reported by 39 stationary detectors within a fixed time interval with a total of 1440 samples. Specifically, we used CSPNs using Poisson leaf nodes and compared them to Poisson SPNs [22]. The task was to predict the next time snapshot ($|\mathbf{Y}| = 39$) from a previous one ($|\mathbf{X}| = 39$). We trained both CSPNs and SPNs controlling the depth of the models. The CSPNs used GLMs with exponential link function as leaf models. The results are summarized in Fig. 3. As one can see CSPNs are more accurate; the root mean squared error (RMSE) is always lower. As expected, deeper models have lower predictive error compared to shallow CSPNs. Moreover, smaller CSPNs perform equally well or

Table 1 Average test conditional log-likelihood (CLL) and test accuracy of the mean field (MF) model, mixture density network (MDN), and neural conditional SPN (CSPN) on multilabel image classification tasks. Predictions on MNIST and Fashion are counted as accurate only if all 16 labels are correct. For CelebA, we report the average accuracy across all labels. The best results are marked in bold. As one can see, the additional representational power of CSPNs yields notable improvements [33]

	CLL			Accuracy		
	MF	MDN	CSPN	MF	MDN	CSPN
MNIST	–0.70	–0.61	**–0.54**	74.1%	76.4%	**78.4%**
Fashion	–0.95	–0.73	**–0.70**	73.4%	73.7%	**75.5%**
CelebA	–12.1	–11.6	**–10.8**	86.6%	85.3%	**87.8%**

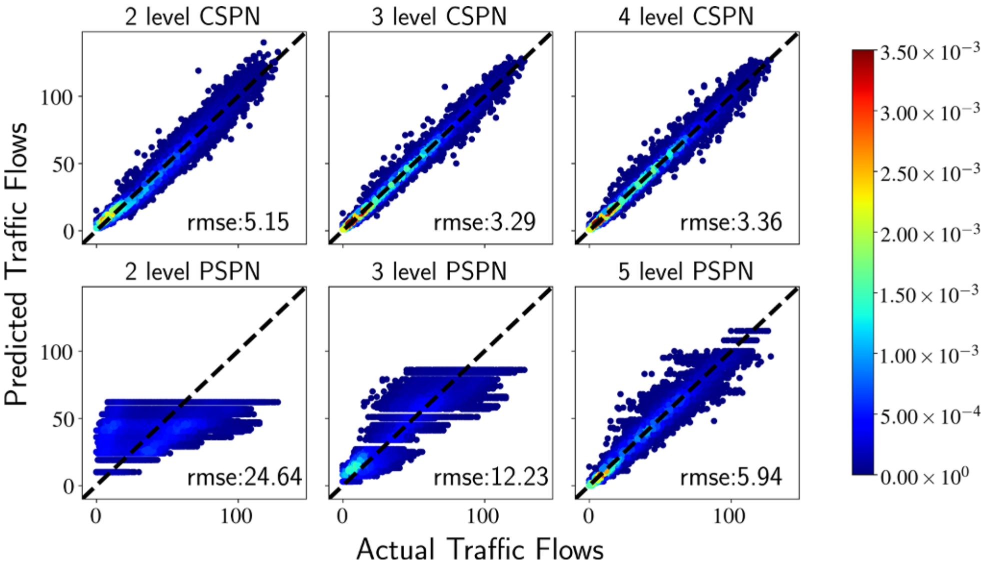

Fig. 3 Comparing traffic flow predictions (RMSE, lower is better) of Poisson CSPNs (*top*) versus SPNs (*bottom*, PSPNs) for shallow (*left*) or deep models (*center and right*). CSPNs are consistently more accurate than corresponding SPNs and, as expected, deeper CSPNs outperform shallow ones (*center and right*). Taken from [33]. (Best viewed in color)

even better than SPNs. This provides clear evidence for the benefit of directly modeling a conditional distribution as well as the expressive power of CSPNs.

To summarize, to be able to build more complex AI models, we have extended the concept of sum-product networks (SPNs) towards conditional distributions by introducing conditional SPNs (CSPNs). Conceptually, they combine simpler models in a hierarchical fashion in order to create a deep representation that can model multivariate and mixed conditional distributions while maintaining tractability. They can be used to impose structure on deep probabilistic models and, in turn, significantly boost their power as demonstrated by our experimental results.

3 Interactively arguing with a classifier

However, CSPNs are deep models and consequently not easy to understand and debug for humans. Therefore, we worked on putting the expert back into the loop. Specifically, we now demonstrate how to constrain the underlying decision logic of deep classifiers by interacting with humans.

To this end, we developed the novel learning setting of explanatory interactive learning (XIL) [38] within CAML. Here, the interaction takes the following form. In each step, the learner explains its interactive query to the user. That is, the machine provides its arguments for its decision. Then, the user responds by proving feedback on the arguments, correcting the prediction and arguments, if necessary. To correct the predictions, one either makes use of automatically generated counterexamples or regularizes the gradients in order to penalize wrong explanations. Recently, we have demonstrated how to make use of influence functions (IFs)—a well known robust statistic [5, 15]—to correct the model's behaviour more effectively.

They trace the model's prediction through the learning algorithm and back to its training data, where the model parameters ultimately derive from, in a closed-form.

Influence Functions. Mathematically, an influence function takes the following form:

$$I(z, z_{\text{test}})^{\text{T}}_{\text{IF}} := -\nabla_{\theta} L(z_{\text{test}}, \widehat{\theta})^{\text{T}} H^{-1}_{\widehat{\theta}} \nabla_{x} \nabla_{\theta} L(z, \widehat{\theta}) \; ,$$

where z and z_{test} are a training sample and a test sample respectively, L denotes the loss, x the input, θ the model parameters and $H := 1/n \sum_{i=1}^{n} \nabla^2_{\theta} L(z_i, \widehat{\theta})$ the Hessian. $I(z, z_{\text{test}})^{\text{T}}_{\text{IF}}$ indicates the most influential direction of perturbing z for z_{test}, and the features of z in this direction explains why the prediction on z_{test} is made. Using just

$$I(z, \theta)^{\text{T}}_{\text{IF}} := H^{-1}_{\widehat{\theta}} \nabla_{x} \nabla_{\theta} L(z, \widehat{\theta})$$

computes the influence of z to θ based on the second-order approximation of the empirical loss around θ. Generally, $H^{-1}_{\widehat{\theta}}$ provides the curvature information of the parameter space and offers a better local approximation of the loss compared to input gradient, and $\nabla_x \nabla_\theta L(z, \widehat{\theta})$ points to the direction in which perturbing the training point z leads to most significant model update. Since we are mainly interested in the latter information, we replace $H^{-1}_{\widehat{\theta}}$ by the identity matrix and, hence, propose the sum of $\nabla_x \nabla_\theta L(z, \widehat{\theta})$ as a more robust statistics for explanatory interactive ML.

To see this, consider Fig. 4. It gives some insights and intuitions on IG-generated explanations and IF $\odot$ IG-generated explanations by visualizing their vector fields and l^2-norm generated by a three-layer MLP on some synthetic 2D classification data sets. As [29] noted, input gradients

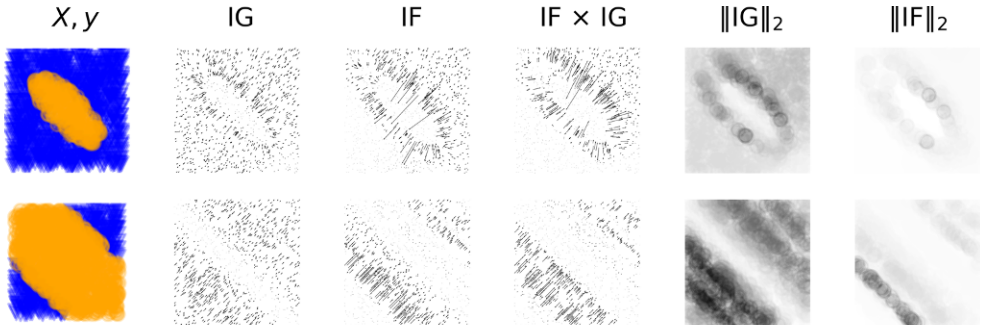

Fig. 4 (Best viewed in color) Input Gradients (IG) versus Influence Function (IF) on two 2D data sets. From *left to right:* data, vector fields of IG and IF as well as their component-wise product, l^2-norm of IG and IF vectors. As one can see, IF integrates the different reasons for a decision into a better explanation

are sometimes noisy and not interpretable on their own. One can see that the vector field of IF $\odot$ IG is sharper around decision boundaries, while IGs yield quite blurry and noisy explanations over the whole domain. Since the decision boundary describes the model's behavior, having a less noisy and ambiguous decision boundary yields a better description of the model.

The "Right for the Better Reasons" Loss. To make use of IFs for explanatory interactive learning, i.e., to argue with the classifier about its decision and reasons for them, we built upon the work on "Right for the Right Reasons"(RRR) [30], we proposed to improve the efficiency by formulating the constraints on the explanations based on the more robust statistic to make the model right for better reasons (RBR). That is, we use the influence function (IF) to compute saliency maps of features and penalize features according to user feedback using standard gradient-based methods.

To this end, we defined the loss function as a weighted sum of the right answer loss (cross-entropy), the right reason loss (user feedback on saliency map) and l^2 regularization:

$$L(\theta, X, y, A) = \underbrace{\frac{1}{N}\sum_{n=1}^{N}\sum_{k=1}^{K} - y_{nk}\log(\widehat{y}_{nk})}_{\text{right answers}}$$

$$+ \underbrace{\lambda_1 \sum_{n=1}^{N}\sum_{d=1}^{D} (A_{nd}\, I(z, \theta)_{\mathrm{IF}}^{\mathrm{T}} \odot I_{\mathrm{IG}})^2}_{\text{right reasons}} + \underbrace{\lambda_2 \sum_{i} \theta_i^2}_{\text{regularization}}$$

where $A_{nd} \in \{-1, 0, 1\}^{n,d}$ encodes user feedback. This loss poses a bias towards the features annotated as -1s, against the features annotated as 1s and ignores the rest. We note that one should be mindful of the faithfulness of the saliency map when formulating right reason loss. This is because plugging in an unfaithful saliency map may lead to non-convergence. And we use the influence of z on the model parameters, $I(z, \theta)_{\mathrm{IF}}^{\mathrm{T}}$, as a measure to approximate the relevance of each feature of z on the model.

RBR results in higher adversarial robustness. We trained an eight-layer MLP as the classifier on the toy color data set from [30] and MNIST [17] by directly constraining IFs. The toy color data set consists of 5∗5 images, and it entails two independent rules: (1) four corner pixels are the same and (2) top middle three pixels are different. Samples satisfying both rules belong to class 1, and samples satisfying neither belong to class 2.

As a baseline, a vanilla classifier trained without any form of constraint and a classifier trained with RRR were used. To generate adversarial examples, we applied the scheme of the Fast Gradient Sign Method (FGSM) [13] but replaced the gradient with the influence function. Fig. 5 shows the accuracy of these three models on the adversarial examples with increasing perturbations. As one can see in Fig. 5, when perturbation increases from 10 to 200 on MNIST, the accuracy of the RBR model dropped by less than 10%, while the vanilla and RRR model dropped by almost 80%. On toy color data set, the accuracy of RBR model barely dropped with increasing perturbation, while the vanilla and RBR model dropped by around 20% and

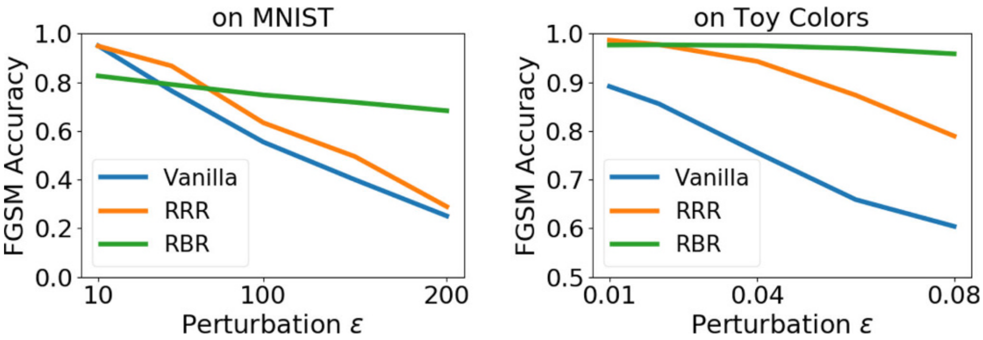

Fig. 5 Accuracy of the vanilla model, RRR and RBR on adversarial examples with increasing perturbations ε

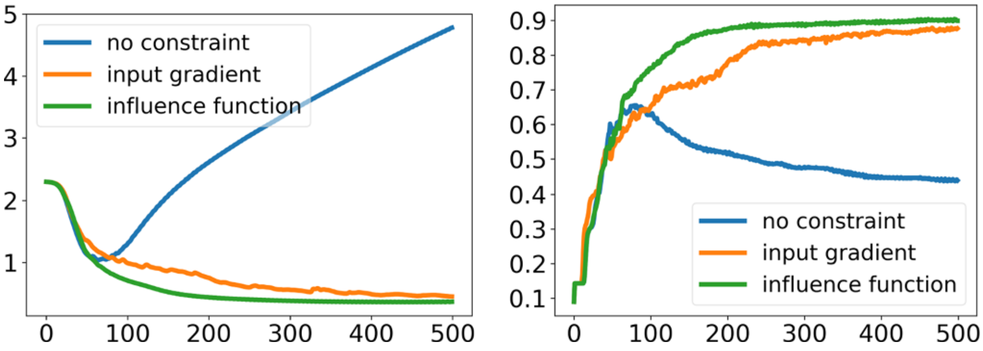

Fig. 6 The cross-entropy (*left image*) and accuracy (*right image*) of the classifier when training on decoyed MNIST with resp. no constraints, IG constraints and IF constraints

30% respectively. This experiment demonstrates that the RBR model is much more robust to adversarial perturbations on both data sets compared to the vanilla and the RRR model.

RBR needs less many iterations. On MNIST, we then trained three MLPs, using no feedback, IG feedback (RRR) and IF feedback (RBR). The cross-entropy and accuracy on the test set reflect how well the model generalizes to unseen data. They are shown over the training epochs in Fig. 6. Without any user feedback, we observed accuracy of 100% on training sets. But on the test set, the cross-entropy is surging and the accuracy dropping to random, suggesting that the model overfits to the confounding factor and does not generalize at all. Providing IF feedback prevents the classifier from learning the confounding rules since the decreasing cross-entropy and improved accuracy on the test set implies the model is able to generalize. Moreover, the convergence speed is much faster compared to RRR.

Arguing with a Deep Network on PASCAL VOC 2007. Finally, we considered the PASCAL VOC 2007 data set [8]. As classifier we used pre-trained VGG-16 [34] and fine-tune it on this data set. PASCAL VOC 2007 consists of labeled images from twenty object classes in realistic scenes, and we reduce the problem to two object classes, horse and dog, due to time restriction. Since there is a class imbalance in the data set, we used the balanced accuracy score defined as the average of recall obtained on each class as an accuracy measure. Without user feedback on the explanations, our fine-tuned vanilla classifier reached accuracy of 99% and 87% on the training set and test set resp.

Now, we started to argue with the classifier. As feedback we encoded the source tag features—a potential confounder—in A to correct the deep network with RBR. Fig. 7a shows an example for user feedback on one instance. The pixels covered by the dark overlay over the image are unsalient features annotated by user feedback, and the rest are not annotated which means they are not explicitly constrained by RBR. In order to investigate the effectiveness of this argumentation-based correction, we also randomized the user-annotated relevant features resp. the irrelevant features across the whole test set. We call the samples with randomized irrelevant features counter samples, and the samples with randomized relevant features as random samples. Fig. 7b and c show a counter sample and a random example. Intuitively, if a classifier is right for the right reasons, the accuracy on the counter examples should be high because the classifier has all the salient features to make decisions, and the accuracy on the random examples should be low as no salient feature is present.

We applied input gradients across the test set to inspect the model's underlying behavior by human perception, and we confirmed that the classifier often accidentally focuses on the source tags to make predictions, as presented in [16].

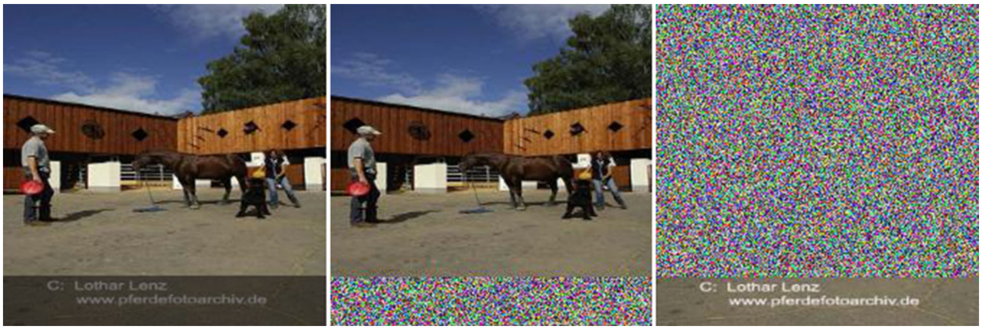

Fig. 7 Left is an original image sample from test set of PASCAL VOC 2007 overlayed with user-annotated mask (the dark overlay denotes 1s in the feedback matrix and the rest are 0s), middle is the corresponding counter example where the user-annotated unsalient features are randomized, right is the corresponding random example where the salient features are randomized

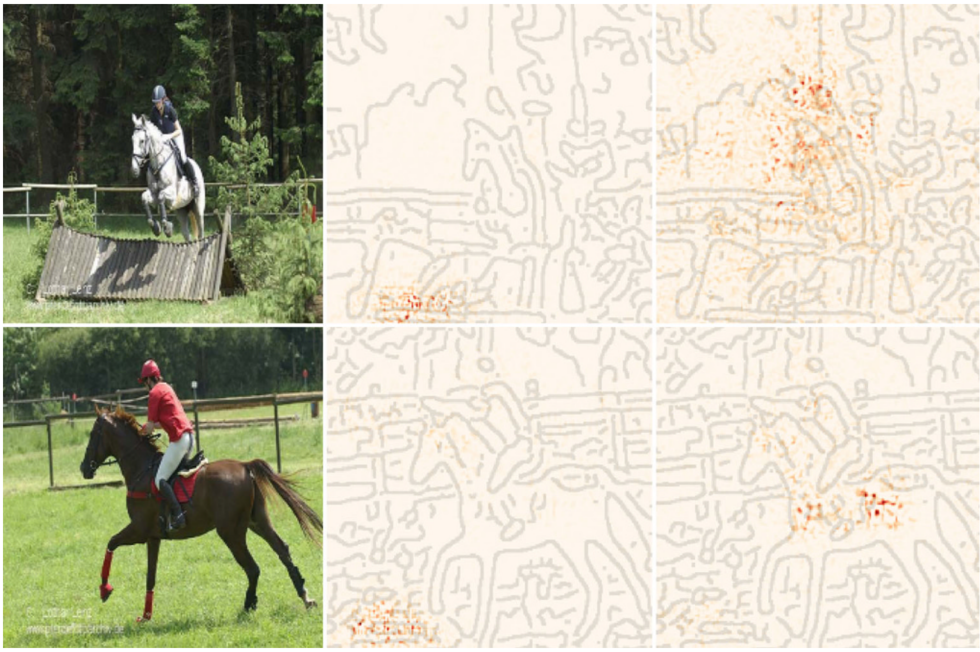

Fig. 8 (Best viewed in color) Revising VGG-16 on PASCAL VOC 2007. Horse images (*left*) and their saliency maps before (*Middle*) and after (*Right*) correction. The saliency maps are overlaid with an edge filtered original image for better interpretability. As one can clearly see, VGG-16 decisions are based on the source tag but can be revised by the user. Before revising, the heatmap hightlights the bottom left corner where the source tags lie as salient regions. This region is no longer salient after correction

Fig. 8 shows some random samples from the test set as well as their saliency maps before and after correction. As one can see, the salient region for the vanilla classifier is mainly on the left bottom corner where the source tags lie. But after the feedback is given, the classifier does not look at the source tags any more and the salient region lies mostly on the target object. Furthermore, without any feedback, the classifier achieved about 75% accuracy on the counter examples, but only about 55% on random examples. This suggests that the classifier did not learn to classify objects and used the confounding factor to classify instead. Fortunately, this unwanted behavior can be corrected by penalizing irrelevant features based on user feedback, and the accuracy for the counter examples dropped to about 53% and the accuracy for the random examples increased to about 63%. This suggests that the classifier learnt to focus on the target object to make decisions.

This confirms the necessity of understanding the behavior of models and also shows clear evidence of the effectiveness of arguing with a model's explanations in high-dimensional image domains.

4 Conclusions

Machine learning and argumentation represent two different solutions for AI. We argue that combining both solutions could bring great benefit. For example, combining deep classifiers with knowledge expressed as arguments allows one to leverage different forms of abstractions within argumentation mining. Argumentation for machine learning can yield argumentation-based learning methods where the machine and the user argue about the learned model with the common goal of providing results of maximum utility to the user. In this paper, we offered an overview of our recent steps towards this combination and in turn towards understanding and arguing with machine learning models. Specifically, We reviewed our recent, efficient regularization by interacting with the explanations of machine learning models to correct them. We illustrated how to do this for differentiable models using influence functions and that this can help to avoid "Clever Hans"-like moments. Besides, as conventional neural function approximators used for predictive tasks are deterministic, and density approximators are in general intractable, we also touched upon our recent work on conditional sum-product networks. This is a deep conditional density approximator which can both maintain the expressive power and a wide range of tractable (conditional) inference routines at the same time.

Acknowledgements We thank all the coauthors of the corresponding papers such as Andrea Galassi, Marco Lippi, Paolo Torroni, Arseny Skryagin, Karl Stelzner, Alejandro Molina, Fabrizio Ventola, Patrick Schramowski, Wolfgang Stammer, Stefano Teso, Anna Brugger, Franziska Herbet, Hans-Georg Luigs, and Anne-Katrin Mahlein, This work was supported by the German Science Foundation project "CAML: Argumentative Machine Learning" as part of the SPP 1999 (RATIO).

Funding Open Access funding provided by Projekt DEAL.

References

1. Aharoni E, Polnarov A, Lavee T, Hershcovich D, Levy R, Rinott R, Gutfreund D, Slonim N (2014) A benchmark dataset for automatic detection of claims and evidence in the context of controversial topics. In: Proceedings of the first workshop on argumentation mining
2. Badgeley MA, Zech JR, Oakden-Rayner L, Glicksberg BS, Liu M, Gale W, McConnell MV, Percha B, Snyder TM, Dudley JT (2019) Deep learning predicts hip fracture using confounding patient and healthcare variables. npj Digit Med. https://doi.org/10.1038/s41746-019-0105-1
3. Baroni P, Caminada M, Giacomin M (2011) An introduction to argumentation semantics. Knowl Eng Rev 26(4):365–410
4. Choi A, Wang R, Darwiche A (2019) On the relative expressiveness of bayesian and neural networks. Int J Approx Reason 113:303–323
5. Cook RD, Weisberg S (1980) Characterizations of an empirical influence function for detecting influential cases in regression. Technometrics 22(4):495–508
6. De Raedt L, Kersting K, Natarajan S, Poole D (2016) Statistical Relational Artificial Intelligence: Logic, Probability, and Computation. Synth Lect Artif Intell Mach Learn 10(2):1–189
7. Ehrlich M, Shields TJ, Almaev T, Amer MR (2016) Facial attributes classification using multi-task representation learning. In: Proc. of the CVPR workshops
8. Everingham M, Van Gool L, Williams CKI, Winn J, Zisserman A (2007) The PASCAL visual object classes challenge 2007 (VOC2007) result
9. Galassi A, Kersting K, Lippi M, Shao X, Torroni P (2020) Neural-symbolic argumentation mining: An argument in favor of deep learning and reasoning. Front Mach Learn AI 2:52. https://doi.org/10.3389/fdata.2019.00052
10. Galassi A, Kersting K, Lippi M, Shao X, Torroni P (2020) Neural-symbolic argumentation mining: an argument in favour of deep learning and reasoning. Front Big Data. https://doi.org/10.3389/fdata.2019.00052
11. Galassi A, Lippi M, Torroni P (2018) Argumentative link prediction using residual networks and multi-objective learning. In: Proceedings of the 5th workshop on argument mining
12. Gens R, Domingos P (2013) Learning the structure of sum-product networks. In: Proc. of ICML
13. Goodfellow I, Shlens J, Szegedy C (2015) Explaining and harnessing adversarial examples. International Conference on Learning Representations, ICLR.
14. Ide C, Hadiji F, Habel L, Molina A, Zaksek T, Schreckenberg M, Kersting K, Wietfeld C (2015) Lte connectivity and vehicular traffic prediction based on machine learning approaches. In: VTC. IEEE
15. Koh PW, Liang P (2017) Understanding black-box predictions via influence functions. In: Proceedings of the 34th international conference on machine learning, vol 70 (JMLR. org)
16. Lapuschkin S, Wäldchen S, Binder A, Montavon G, Samek W, Müller KR (2019) Unmasking clever hans predictors and assessing what machines really learn. Nat Commun. https://doi.org/10.1038/s41467-019-08987-4
17. LeCun Y, Cortes C, Burges CJ (2010) MNIST handwritten digit database. http://yann.lecun.com/exdb/mnist
18. Lippi M, Torroni P (2015) Context-independent claim detection for argument mining. Twenty-Fourth International Joint Conference on Artificial Intelligence.
19. Lippi M, Torroni P (2016) Argumentation mining: State of the art and emerging trends. ACM Trans Internet Technol 16(2):1–25
20. Manhaeve R, Dumancic S, Kimmig A, Demeester T, Raedt LD (2018) Deepproblog: neural probabilistic logic programming. In: Proc. of NeurIPS 2018, pp 3753–3763
21. McCullagh P (1984) Generalized linear models. EJOR. https://doi.org/10.1007/978-1-4899-3242-6
22. Molina A, Natarajan S, Kersting K (2017) Poisson sum-product networks: a deep architecture for tractable multivariate poissons. In: Proc. of AAAI
23. Mozina M, Guid M, Krivec J, Sadikov A, Bratko I (2008) Fighting knowledge acquisition bottleneck with argument based machine learning. In: Proceedings of the 18th European Conference on Artificial Intelligence (ECAI), pp 234–238
24. Niculae V, Park J, Cardie C (2017) Argument mining with structured svms and rnns (arXiv preprint arXiv:1704.06869)
25. Obermeyer Z, Powers B, Vogeli C, Mullainathan S (2019) Dissecting racial bias in an algorithm used to manage the health of populations. Science 366(6464):447–453
26. Poon H, Domingos P (2011) Sum-product networks: a new deep architecture. In: Proc. of UAI
27. Rinott R, Dankin L, Alzate C, Khapra MM, Aharoni E, Slonim N (2015) Show me your evidence-an automatic method for context dependent evidence detection. In: Proceedings of the conference on empirical methods in natural language processing
28. Riveret R, Gao Y, Governatori G, Rotolo A, Pitt J, Sartor G (2019) A probabilistic argumentation framework for reinforcement learning agents - towards a mentalistic approach to agent profiles. Auton Agent Multi Agent Syst 33(1–2):216–274
29. Ross AS, Doshi-Velez F (2018) Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients. In: Thirty-second AAAI conference on artificial intelligence (AAAI)
30. Ross AS, Hughes MC, Doshi-Velez F (2017) Right for the right reasons: training differentiable models by constraining their explanations. In: Proceedings of the twenty-sixth international joint conference on artificial intelligence, IJCAI-17
31. Schramowski P, Stammer W, Teso S, Brugger A, Herbet F, Shao X, Luigs HG, Mahlein AK, Kersting K (2020) Right for the wrong scientific reasons: revising deep networks by interacting with their explanations (arXiv preprint arXiv:2001.05371)
32. Sebeok TA, Rosenthal RE (1981) The clever hans phenomenon: Communication with horses, whales, apes, and people. Ann NY Acad Sci. https://doi.org/10.1111/j.1749-6632.1981.tb34458.x
33. Shao X, Molina A, Vergari A, Stelzner K, Peharz R, Liebig T, Kersting K (2019) Conditional sum-product networks: imposing structure on deep probabilistic architectures. In: ICML 2019 workshop on tractable probabilistic models
34. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition (arXiv preprin)
35. Skryagin A, Stelzner K, Molina A, Ventola F, Kersting K (2020) Splog: Sum-product logic. In: Proceedings of the 2nd international conference on probabilistic programming
36. Strobl EV, Zhang K, Visweswaran S (2019) Approximate kernel-based conditional independence tests for fast non-parametric causal discovery. J Causal Inference. https://doi.org/10.1515/jci-2018-0017
37. Teso S, Kersting K (2019) Explanatory interactive machine learning. In: Proceedings of the 2nd AAAI/ACM Conference on AI, Ethics, and Society (AIES)

38. Teso S, Kersting K (2019) Explanatory interactive machine learning. Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society, pp 239–245
39. Thimm M, Kersting K (2017) Towards argumentation-based classification. In: In working notes of the IJCAI workshop on logical foundations of uncertainty and machine learning