Explanatory Interactive Machine Learning

Stefano Teso
Department of Computer Science
KU Leuven, Belgium
stefano.teso@cs.kuleuven.be

Kristian Kersting
Department of Computer Science and
Centre for Cognitive Science
TU Darmstadt, Germany
kersting@cs.tu-darmstadt.de

Abstract

Although interactive learning puts the user into the loop, the learner remains mostly a black box for the user. Understanding the reasons behind predictions and queries is important when assessing how the learner works and, in turn, trust. Consequently, we propose the novel framework of *explanatory* interactive learning where, in each step, the learner explains its query to the user, and the user interacts by both answering the query and correcting the explanation. We demonstrate that this can boost the predictive and explanatory powers of, and the trust into, the learned model, using text (e.g. SVMs) and image classification (e.g. neural networks) experiments as well as a user study.

Introduction

Trust lies at the foundation of major theories of interpersonal relationships in psychology (Simpson 2007). In particular, Hoffman and others (2013) argue that interpersonal trust depends on the "perceived competence, benevolence (or malevolence), understandability, and directability—the degree to which the trustor can rapidly assert control or influence when something goes wrong." They and others (Waytz and others 2014; Wang and others 2016) also show that trust into machines follows similar patterns, with some notable differences: it is often inappropriate to attribute benevolence/malevolence to machines, and trust into machines suffers from different biases than trust into individuals. The differences, however, do not affect the argument that interaction and understandability are central to trust in machine learners, too. The *competence* of a classifier can be assessed by monitoring its behavior and beliefs over time, *directability* can be achieved by allowing the user to actively teach the model how to act and what to believe, while *understandability* can be approached by explaining the model's decisions.

Surprisingly, the link between interacting, explaining and building trust has been largely ignored by the machine learning literature. On one hand, existing approaches focus on passive learning only, and do not consider interaction between the user and the learner (Bucilu and others 2006; Ribeiro and others 2016; Lundberg and others 2016). On the other hand, interactive learning frameworks such as active (Settles 2012) and coactive learning (Shivaswamy and others 2015) do not consider the issue of trust. In active learning, for instance, the model presents unlabelled instances to a user, and in exchange obtains their label. This is completely opaque: the user is oblivious to the model's beliefs and reasons for predictions and to how they change in time, and cannot see the consequences of her own instructions. In coactive learning, the user sees and corrects the system's prediction, if necessary, but the predictions are not explained to her. So, why should users trust models learned interactively?

To fill this gap, we propose the novel framework of *explanatory interactive learning* (XIL). Here the interaction takes the following form. In each step, the learner explains its interactive query to the user, and she responds by correcting the prediction and explanations, if necessary, to provide feedback. We also present a model-agnostic method, called CAIPI, instantiating our framework for active learning. CAIPI extends active learning in several ways. Akin to coactive learning (Shivaswamy and others 2015), query instances are accompanied by the the model's corresponding *predictions*. This allows the user to check whether the model is right or wrong on the chosen instance. However, nothing prevents the model from being right (or wrong) for the wrong reasons, e.g., when there are ambiguities in the data such as confounders (Ross and others 2017). To avoid this issue, CAIPI accompanies predictions with corresponding *explanations*, computed by any local explainer of choice (Ribeiro and others 2016; Lundberg and others 2016; Ross and others 2017; Ribeiro and others 2018); in this paper we use LIME[1] (Ribeiro and others 2016), a simple model-agnostic explainer that allows to easily compute explanations and present them to the user as interpretable (visual) artifacts. By witnessing the evolution of the explanations—like a teacher supervising the progress of a student—the user can see whether the model eventually "gets it". Finally, the user can even correct the explanation presented to guide the learner. This *correction* step is crucial for more directly affecting the learner's beliefs and is integral to modulating trust (Hoffman and others 2013; Kulesza and others 2015). Explanation corrections also facilitate learning (the right concept), especially in problematic cases that labels alone can not handle (Ross and others

[1] CAIPIrinhas are made out of LIMEs.

2017), as shown by our experiments. Overall, CAIPI is the first active learning approach that employs explanation corrections as an additional feedback channel in a model- and explainer-agnostic fashion. Our empirical evidence demonstrates that this interaction through explanations can modulate trust and booat the effectiveness of learning, also compared to state-of-the-art.

We proceed as follows. First, we touch upon additional related work. Then we introduce XIL and derive CAIPI. Before concluding, we present our empirical evaluation.

Further Related Work

Machine learning explainers fall in two classes. Global approaches aim to explain a black-box model by converting it as a whole to a more interpretable format (Bucilu and others 2006; Bastani and others 2017). Local approaches instead interpret individual predictions (Lundberg and others 2016). Surprisingly, they do not consider interaction between the user and the model. Existing interactive learning approaches such as active, coactive, and active imitation learning do not consider the issue of explanations and trust, as already discussed. Given the centrality of the user in recommendation, interactive preference elicitation makea use of conversational interaction to improve trust and directability (Peintner and others 2008; Chen and others 2012), but often rely on rudimental learning strategies (if any).

Indeed, learning from explanations has been explored in concept learning (Mitchell and others 1986; DeJong and others 2011) and probabilistc logic programming (Kimmig and others 2007), where explanations are themselves logical objects. Unfortunately, these results are tied to logic-based models and make use of rather opaque forms of explanations (e.g. logic proofs), which can be difficult to grasp for non-experts. Explanatory interactive learning instead leverages explanations for mainstream machine learning approaches.

More recently, researchers explored feature supervision (Raghavan and others 2006; 2007; Druck and others 2008; 2009; Settles 2011; Attenberg and others 2010) and rationales (Zaidan and others 2007; 2008; Sharma and others 2015), which leverage both label- and feature-level (or sentence-level, for rationales) supervision with the aim of improving learning efficiency. These works show that providing rationales, even from scratch, can be easy for human annotators (Zaidan and others 2007), sometimes even more so than providing the labels themselves (Raghavan and others 2006). These approaches assume that the learner and the user can both work with the same features, which is not the case in general. More generally, most of these approaches are black-box, i.e., they do produce no explanations at all. The connection to directability and trust is not explicitly made. Those approaches that do are either model- or application specific (e.g. (Zaidan and others 2007)). Explanatory interactive learning generalizes these ideas to arbitrary classification tasks and models. We remark that the feedback techniques proposed in these works are orthogonal to explanatory interactions and can be easily combined with it. We showcase this in one of our experiments, which leverages the corrective technique of (Zaidan and others 2007).

Finally, the UI community also investigated meaningful interaction strategies so that the user can build a mental model of the system. In (Stumpf and others 2009) the user is allowed to provide explanations, while (Kulesza and others 2015) provides an explanation-centric approach to interactive teaching. These works however focus on simple machine learning models, like Naïve Bayes, while explanatory interactive learning is much more general.

Explanatory Interactive Learning (XIL)

In XIL, a learner is able to interactively query the user (or some other information source) to obtain the desired outputs at data points. The interaction takes the following form. At each step, the learner considers a data point (labeled or unlabeled), predicts a label, and provides explanations of its prediction. The user responds by correcting the learner if necessary, providing a slightly improved—but not necessarily optimal—feedback to the learner.

Let us now instantiate this schema to *explanatory active learning*—combining active learning with local explainers. Indeed, other interactive learning can be made explanatory too, including coactive learning (Shivaswamy and others 2015), active imitation learning (Judah and others 2012), and mixed-initiative interactive learning (Cakmak and others 2011), but this is beyond the scope of this paper.

Active learning. The active learning paradigm targets scenarios where obtaining supervision has a non-negligible cost. Here we cover the basics of pool-based active learning, and refer the reader to two excellent surveys (Settles 2012; Hanneke and others 2014) for more details. Let $\mathcal{X}$ be the space of instances and $\mathcal{Y}$ be the set of labels (e.g. $\mathcal{Y} = \{\pm 1\}$). Initially, the learner has access to a small set of labelled examples $\mathcal{L} \subseteq \mathcal{X} \times \mathcal{Y}$ and a large pool of unlabelled instances $\mathcal{U} \subseteq \mathcal{X}$. The learner is allowed to query the label of unlabelled instances (by paying a certain cost) to a user functioning as annotator, often a human expert. Once acquired, the labelled examples are added to $\mathcal{L}$ and used to update the model. The overall goal is to maximize the model quality while keeping the number of queries or the total cost at a minimum. To this end, the query instances are chosen to be as informative as possible, typically by maximizing some informativeness criterion, such as the expected model improvement (Roy and others 2001) or practical approximations thereof. By carefully selecting the instances to be labelled, active learning can enjoy much better sample complexity than passive learning (Castro and others 2006; Balcan and others 2010). Prototypical active learners include max-margin (Tong and Koller 2001) and Bayesian approaches (Krause and others 2007); recently, deep variants have been proposed (Gal and others 2017).

However, active—showing query data points—and even coactive learning—showing additionally the prediction of the query data point— do not establish trust: informative selection strategies just pick instances where the model is uncertain and likely wrong. Thus, there is a trade-off between query informativeness and user "satisfaction," as noticed and explored in (Schnabel and others 2018). In order to properly modulate trust into the model, we argue it is essential to present explanations.

Local explainers. There are two main strategies for interpreting machine learning models. Global approaches aim to explain the model by converting it *as a whole* to a more interpretable format (Bucilu and others 2006; Bastani and others 2017). Local explainers instead focus on the arguably more approachable task of explaining *individual predictions* (Lundberg and others 2016). While explainable interactive learning can accommodate any local explainer, in our implementation we use LIME (Ribeiro and others 2016), described next[2]. The idea of LIME (Local Interpretable Model-agnostic Explanations) is simple: even though a classifier may rely on many uninterpretable features, its decision surface around any given instance can be locally approximated by a simple, interpretable *local model*. In LIME, the local model is defined in terms of simple features encoding the presence or absence of *basic components*, such as words in a document or objects in a picture[3]. An explanation can be readily extracted from such a model by reading off the contributions of the various components to the target prediction and translating them to an interpretable visual artifact. For instance, in document classification one may highlight the words that support (or contradict) the predicted class.

Formally, let $f : \mathcal{X} \to \mathcal{Y}$ be an uninterpretable classifier (e.g., a dense linear model, a random forest, a deep network), $\hat{y} = f(x)$ the target prediction, and for each basic component i let $\psi_i(x)$ be the corresponding indicator function. In order to explain the prediction, LIME produces an interpretable model $g : \mathcal{X} \to \mathcal{Y}$, based solely on the interpretable features $\{\psi_i\}_i$, that approximates f in the neighborhood of x. Here g can be any sufficiently interpretable model, for instance a sparse linear classifier or a shallow decision tree. Computing g amounts to solving $\operatorname{argmin}_g \ \ell_x(f, g) + \Omega(g)$, where ℓ_x is a "local loss" that measures the *fidelity* of g to f in the neighborhood of x, and $\Omega(g)$ is a regularization term that controls the complexity and interpretability of g.

For the sake of simplicity, we focus on LIME in conjunction with sparse linear models of the form $g(x) = \langle \boldsymbol{w}, \boldsymbol{\psi}(x) \rangle + b$, where $\langle \cdot, \cdot \rangle$ denotes the dot product. In order to enhance interpretability, at most k non-zero coefficients are allowed, where k is sufficiently small (see our Empirical Analysis for the values we use). Specifically, LIME measures the fidelity of the linear approximation with a "local" L_2 distance, namely $\ell_x(f, g) = \int_{x'} k(x, x')(f(x') - g(x'))^2 dx'$. In practice, this problem is solved by approximating the integral as a sum over a large enough set $\mathcal{S} \subseteq \mathcal{X}$ of instances sampled uniformly at random[4] and solving the sparsity-constrained least-squares problem: $g = \operatorname{argmin}_g \sum_{x' \in \mathcal{S}} k(x, x')(f(x') - g(x'))^2 \quad \text{s.t. } \|\boldsymbol{w}\|_0 \leq k$. Note that g does depend on both the target instance x and on the prediction $\hat{y} = f(x)$. The relevance and polarity of all components can be readily read off from the weights $\boldsymbol{w}$: $|w_j| > 0$ suggests that the jth component does contribute to the overall prediction, while $w_j > 0$ and $w_j < 0$ imply that, when present, the jth component drives the prediction toward $\hat{y}$ or away from it, respectively. Finally, this information is used to construct a (visual) explanation.

Algorithm 1 CAIPI takes as input a set of labelled examples $\mathcal{L}$, a set of unlabelled instances $\mathcal{U}$, and iteration budget T.

$f \leftarrow \text{FIT}(\mathcal{L})$
repeat
  $x \leftarrow \text{SELECTQUERY}(f, \mathcal{U})$
  $\hat{y} \leftarrow f(x)$
  $\hat{z} \leftarrow \text{EXPLAIN}(f, x, \hat{y})$
  Present x, $\hat{y}$, and $\hat{z}$ to the user
  Obtain y and explanation correction $\mathcal{C}$
  $\{(\bar{x}_i, \bar{y}_i)\}_{i=1}^{c} \leftarrow \text{TOCOUNTEREXAMPLES}(\mathcal{C})$
  $\mathcal{L} \leftarrow \mathcal{L} \cup \{(x, y)\} \cup \{(\bar{x}_i, \bar{y}_i)\}_{i=1}^{c}$
  $\mathcal{U} \leftarrow \mathcal{U} \setminus (\{x\} \cup \{\bar{x}_i\}_{i=1}^{c})$
  $f \leftarrow \text{FIT}(\mathcal{L})$
until budget T is exhausted or f is good enough
return f

Explanatory Active Learning. Now, we have everything together for explanatory *active* learning and CAIPI. Specifically, we require black-box access to an active learner and an explainer. We assume that the active learner provides a procedure $\text{SELECTQUERY}(f, \mathcal{U})$ for selecting an informative instance $x \in \mathcal{U}$ based on the current model f, and a procedure $\text{FIT}(\mathcal{L})$ for fitting a new model (or update the current model) on the examples in $\mathcal{L}$. The explainer is assumed to provide a procedure $\text{EXPLAIN}(f, x, \hat{y})$ for explaining a particular prediction $\hat{y} = f(x)$. The framework is intended to work for any reasonable learner and explainer.

When using LIME for computing an interpretable model locally around the queries in order to visualize explanations for current predictions, this results in CAIPI as summarized in Alg. 1. At each iteration $t = 1, \ldots, T$ an instance $x \in \mathcal{U}$ is chosen using the query selection strategy implemented by the SELECTQUERY procedure. Then its label $\hat{y}$ is predicted using the current model f, and EXPLAIN is used to produce an explanation $\hat{z}$ of the prediction. The triple $(x, \hat{y}, \hat{z})$ is presented to the user as a (visual) artifact. The user checks the prediction and the explanation for correctness, and provides the required feedback. Upon receiving the feedback, the system updates $\mathcal{U}$ and $\mathcal{L}$ accordingly and re-fits the model. The loop terminates when the iteration budget T is reached or the model is good enough.

During interactions between the system and the user, three cases can occur: **(1) Right for the right reasons:** The prediction and the explanation are both correct. No feedback is requested. **(2) Wrong for the wrong reasons:** The prediction is wrong. As in active learning, we ask the user to provide the correct label. The explanation is also necessarily wrong, but we currently do not require the user to act on it. **(3) Right for the wrong reasons:** The prediction is correct but the explanation is wrong. We ask the user to provide an explanation *correction* $\mathcal{C}$.

[2]We use LIME for simplicity. RRR (Ross and others 2017) and ANCHORS (Ribeiro and others 2018) are valid alternatives.

[3]While not all problems admit explanations in terms of elementary components, many of them do (Ribeiro and others 2016); in this case, LIME assumes these to be provided in advance.

[4]In LIME the samples are taken from the image of ψ, i.e., $\{\psi(x) : x \in \mathcal{X}\}$, and then mapped back to $\mathcal{X}$ to compute their predicted class. We omit this detail for clarity.

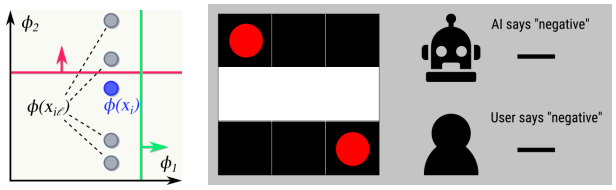

Figure 1: (Left) Mathematical intuition for the counterexample strategy. (Right) Example training round as presented in the questionnaire. The classification is correct but the explanation shows that the two most relevant pixels do not match the true classification rule (as in S3). (Best viewed in color).

The "right for the wrong reasons" case is novel in active learning, and we propose *explanation corrections* to deal with it. They can assume different meanings depending on whether the focus is on component relevance, polarity, or relative importance (ranking), among others. In our experiments we ask the annotator to indicate the components that have been wrongly identified by the explanation as relevant, that is, $\mathcal{C} = \{j : |w_j| > 0 \wedge$ the user believes the jth component to be irrelevant$\}$. In document classification, $\mathcal{C}$ would be the set of words that are irrelevant according to the user but relevant for the model.

Given the correction $\mathcal{C}$, we are faced with the problem of explaining it back to the learner. We propose a simple strategy to achieve this. This strategy is embodied by ToCounterExamples. It converts $\mathcal{C}$ to a set of *counterexamples* that teach the learner not to depend on the irrelevant components. In particular, for every $j \in \mathcal{C}$ we generate c examples $(\bar{x}_1, \bar{y}_1), \ldots, (\bar{x}_c, \bar{y}_c)$, where c is an application-specific constant. Here, the labels $\bar{y}_i$ are identical to the prediction $\hat{y}$. The instances $\bar{x}_i$, $i = 1, \ldots, c$ are also identical to the query x, except that the jth component (i.e. $\psi_j(x)$) has been either randomized, changed to an alternative value, or substituted with the value of the jth component appearing in other training examples of the same class. In sudoku, each $\bar{x}_i$ would be a copy of the query sudoku x where the cells in $\mathcal{C}$ have been (for instance) filled with random numbers consistent with the predicted label. This process produces $c|\mathcal{C}|$ counterexamples, which are added to $\mathcal{L}$.

Why is this data augmentation a sensible idea? To see this, consider the case of linear max-margin classifiers. Let $f(x) = \langle \boldsymbol{w}, \boldsymbol{\phi}(x) \rangle + b$ be a linear classifier over two features, ϕ_1 and ϕ_2, of which only the first is relevant. Fig. 1 (left) shows that $f(x)$ (red line) uses ϕ_2 to correctly classify a negative example x_i. In order to obtain a better model (e.g. the green line), the simplest solution would be to enforce an orthogonality constraint $\langle \boldsymbol{w}, (0,1)^\top \rangle = 0$ during learning. Counterexamples follow the same principle. In the separable case, the counterexamples $\{\bar{x}_{i\ell}\}_{\ell=1}^{c}$ amount to additional max-margin constraints (Cortes and others 1995) of the form $y_i \langle \boldsymbol{w}, \boldsymbol{\phi}(\bar{x}_{i\ell}) \rangle \geq 1$. The only ones that influence the model are those on the margin, for which strict equality holds. For all pairs of such counterexamples ℓ, ℓ' it holds that $\langle \boldsymbol{w}, \boldsymbol{\phi}(\bar{x}_{i\ell}) \rangle = \langle \boldsymbol{w}, \boldsymbol{\phi}(\bar{x}_{i\ell'}) \rangle$, or equivalently $\langle \boldsymbol{w}, \boldsymbol{\delta}_{i\ell} - \boldsymbol{\delta}_{i\ell'} \rangle = 0$, where $\boldsymbol{\delta}_{i\ell} = \boldsymbol{\phi}(\bar{x}_{i\ell}) - \boldsymbol{\phi}(x_i)$. In other words, the counterexamples encurage orthogonality between $\boldsymbol{w}$ and the correction vectors $\boldsymbol{\delta}_{i\ell} - \boldsymbol{\delta}_{i\ell'}$, thus approximating the orthogonality constraint above.

	Q1	Q2	Q3
S1	64.7%	35.3%	82.4%
S2	76.5%	64.7%	70.6%
S3	29.4%	11.8%	41.2%

	no corr.	Counterexamples $c=1$	$c=3$	$c=5$	Input Gradients
Train	**0.978**	0.938	0.922	0.924	0.898
Test	0.482	0.821	0.851	**0.858**	0.853

Table 1: Explanatory feedback can boost trust and performance. (Top) User study: percentage of "yes" answers. (Bottom) Accuracy on the fashion MNIST dataset of an MLP without corrections (no corr.), with our counterexample corrections using varying c (middle), and with input gradient constraints (Ross and others 2017).

Most importantly, this data augmentation procedure is model-agnostic, although alternatives indeed exist: Contrastive examples (Zaidan and others 2007), feature ranking (Small and others 2011) for SVMs and constraints on the input gradients for differentiable models (Ross and others 2017). These may be more effective in practice, and CAIPI can accomodate all of them. However, since our strategy is both model- and explainer-agnostic, in the remainder we will stick to it for maximum generality.

Empirical Evidence

Our intention here is to address empirically the following questions: (**RQ1**) Can explanations (and their consistency over time) appropriately modulate the user's trust into the model? (**RQ2**) Can explanation corrections lead to better models? (**RQ3**) Do the explanations necessarily improve as the learner obtains more labels? (**RQ4**) Does the magnitude of this effect depend on the specific learner?

(RQ1) User study. We designed a questionnaire about a machine that learns a simple concept by querying labels (but *not* explanation corrections) to an annotator. The questionnaire, available in the Supplementary Material, was administered to 17 randomly selected undergraduate students from an introductory course on deep learning. We designed a toy binary classification problem (inspired by (Ross and others 2017)) about classifying small (3×3) black-and-white images. The subjects were told that an image is positive if the two top corners are white and negative otherwise. Then they were shown three learning sessions consisting of five query/feedback rounds each. In session 1 (**S1**) every round included the images chosen by the model, the corresponding prediction, and the label provided by a knowledgeable annotator. No explanations were shown. The predictions are wrong for the first three rounds and correct in the last two. Sessions 2 and 3 (**S2**, **S3**) were identical to S1, meaning that at every round *the same example, prediction and feedback label* were shown, but now explanations were also provided. The explanations highlighted the two most relevant pixels, as in Fig. 1 (right). In S2 the explanations converged to the correct rule—they highlight the two top corners—from the

fourth round onwards, while in S3 they did not. Removing the explanations reduces both S2 and S3 to S1. After each session, the subjects were asked three questions: (**Q1**) "Do you believe that the AI system eventually learned to classify images correctly?" (**Q2**) "Do you believe that the system eventually learned the correct classification rule?" (**Q3**) "Would you like to further assess the system by checking whether it classifies 10 random images correctly?" The first two questions test the subject's uncertainty in the predictive ability and beliefs of the classifier, resp., while the last one tests the relationship between predictive accuracy (but *not* explanation correctness) and expected uncertainty reduction. The percentage of "yes" answers is down in Tab. 1(top).

As expected, the uncertainty in the model's correctness depends heavily on what information channels are enabled. When no explanations are shown (S1), only 35% of the subjects assert to believe that the model learned the correct rule (Q2). This percentage almost doubles (65%) when explanations are shown and converge to the correct rule (S2). The need to see more examples also lowers from 82% to 71%, but does not drop to zero. This reflects the fact that five rounds are not enough to reduce the subject's uncertainty to low enough levels. The percentage of subjects asserting that the classifier produces correct predictions (regardless of the learned rule, Q1) also increases from 65% to 77% when correct explanations are shown (S2). When the explanations do not converge (S3), the trend is reversed: Q1 drops to 29% and Q2 to 12%, i.e., most subjects do not believe that the model's behavior and beliefs are in any way correct. This is the only setting where Q3 drops below 50% (41%): witnessing that the model's beliefs do not match the target rule induces distrust (with high certainty). This confirms the previous finding that trust into machines drops when wrong behavior is witnessed (Hoffman and others 2013). Thus, **RQ1** can be answered affirmatively: augmenting interaction with explanations does appropriately drive trust into the model.

Next to the user study, we considered simulated users—as it is common for active learning— to investigate **(RQ2–4)**. To this aim, we implemented CAIPI on top of several standard active learners and applied it to different learning tasks. Note that our goal here is to evaluate the contribution of explanation feedback, not the learners themselves. Indeed, CAIPI can trivially accommodate more advanced models than the ones employed here. In all cases, the model's explanations are computed with LIME [5]. As is common in active learning, we simulate a human annotator that provides correct labels. Explanation corrections are also assumed to be correct and complete (i.e. they identify all false positive components), for simplicity[6]. The specifics of the correction strategy are described in the next paragraphs. Our experimental setup is available at `https://github.com/stefanoteso/caipi`

[5]Due to sampling, LIME may output different explanations for the same prediction. To reduce variance, we ran it 10 times and kept the k components identified most often.

[6]In practice corrections may be incomplete or noisy, especially when dealing with non-experts. This can be handled by, e.g., down-weighting the counterexamples.

(RQ2) Evaluation on a passive setting. We applied our data augmentation strategy to a decoy variant of fashion MNIST, a fashion product recognition dataset[7]. The dataset includes 70,000 images over 10 classes. All images were corrupted by introducing confounders, that is, 4×4 patches of pixels in randomly chosen corners whose shade is a function of the label in the training set and random in the test set (see (Ross and others 2017) for details). The average test set accuracy of a multilayer perceptron (with the same hyperparameters as in (Ross and others 2017)) is reported in Tab. 1 (bottom) for three correction strategies: no corrections, our counterexample strategy (CE), and the input-gradient constraints proposed by (Ross and others 2017) (IG). For CE, for every training image we added $c = 1, 3, 5$ counterexamples where the decoy pixels are randomized. When no corrections are given, the accuracy on the test set is 48%: the confounders completely fool the network. Providing even a single counterexample increases the accuracy to 82%, i.e., the effect of confounders drops drastically. With more counterexamples the accuracy passes the one of IG (85%). This shows that (**RQ2**) counterexamples—and therefore explanation corrections—are an effective measure for improving the model in terms of both predictive performance and beliefs.

(RQ3,4) Actively choosing among concepts and comparison to SOTA. We applied CAIPI to the "colors" dataset of (Ross and others 2017). The goal is to classify 5×5 images with four possible colors. An image is positive if either the four corner pixels have the same color (rule 0) or the three top middle pixels have different colors (rule 1). Crucially, the dataset only includes images where either both rules hold or neither does, that is, labels alone can not disambiguate between the two rules. Explanations highlight the k most relevant pixels, and corrections indicate the pixels that are wrongly identified as relevant. In the counterexamples, the wrongly identified pixels are recolored using all possible alternative colors consistent with $\hat{y}$ [8]. The features are of the form "pixel i has the same color as pixel j" for all $i, j = 1, \ldots, 25$, $i < j$. In this space, the rules can be represented by sparse hyperplanes. We select each rule in turn and provide corrections according to it, and then check whether the feedback drives the classifier toward it. k was set to 4 for rule 0 and to 3 for rule 1. We followed a 10-fold CV strategy.

In a first step, we considered a standard L_2 SVM active learner with the closest-to-margin query selection heuristic (Settles 2012). This classifier can in principle represent both rules, but it is not suited for learning sparse concepts. Indeed, the SVM struggles to learn both rules, and the counterexamples have little effect on it (see the Supplementary Material for the complete results). This is plausible since the L_2 norm cannot capture the underlying sparse concept: even though corrections try to drive the model toward it, the L_2 SVM can still learn *both* rules (as shown by the coefficient curves) without a problem. In other words, the model is not constrained enough.

An L_1 SVM, an active learner tailored for sparse con-

[7]`https://github.com/zalandoresearch/fashion-mnist`

[8]We always discard counterexamples that appear in the test set.

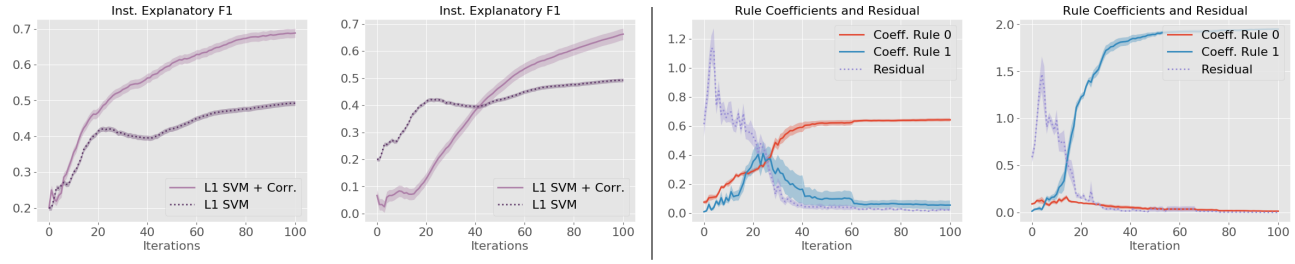

Figure 2: Explanatory feedback ('+ Corr.') can drive an active L_1 SVM towards the right concept (colors problem). (Left) instantaneous F_1 score of the LIME explanations for rule 0 (leftmost) and rule 1 (left middle). (Right) Decomposition of the learned weight vector when the corrections push toward rule 0 (right middle) and rule 1 (rightmost). (Best viewed in color)

Figure 3: Explanatory feedback ('+ Corr.') can boost an active logistic regression (LR) on 20 newsgroups.

cepts (Zhu and others 2004), fares much better. Our results show that the rules greatly benefit this model. To evaluate their effect, we compute the average instantaneous F_1 score of the pixels identified by LIME w.r.t. the pixels truly relevant for the selected rule. This measures the quality of the explanations presented to the user. In addition, we measure the objective quality of the model by decomposing the learned weights using least-squares as $\boldsymbol{w} = \alpha_0 \boldsymbol{w}_0^* + \alpha_1 \boldsymbol{w}_1^* +$ residual, where $\boldsymbol{w}_i^*$ is the "perfect" weight vector of rule $i = 0, 1$. The instantaneous F_1 and change in coefficients can be viewed in Fig. 2. Now that the model can capture the target concepts, the contribution of counterexamples is very noticeable: the L_1 SVM is biased toward rule 1, as it is sparser (data not shown), but it veers clearly toward rule 0 when corrections are provided and learns rule 1 faster when corrections push toward it. These results show clearly that explanation feedback can drive the classifier toward the right concept, so long as the chosen model can capture it clearly.

(RQ3,4) Active learning for text classification. Finally, we applied CAIPI to distinguishing between "Atheism" and "Christian" posts in the 20 newsgroups dataset using logistic regression with uncertainty sampling. Headers and footers were removed; only adjectives, adverbs, nouns, and verbs were kept and stemmed. As gold standard for the explanations, we selected $\approx \frac{1}{5}$ of the words as relevant using feature selection. Here the LIME-provided explanations identified the k most relevant words, while corrections identified the falsely relevant words. For each document, k was set to the number of truly relevant words. To showcase CAIPI's flexibility, the counterexamples were generated with the strategy proposed in (Zaidan and others 2007), adapted to produce feedback based on the falsely relevant words only. The 10-fold cross-validated results can be found in Fig. 3. The plots show that the model with explanation corrections is steadily better in terms of explanation quality—over the test set (left) and queries (right)—than the baseline without corrections. The predictive performance can be found in the longer version of the paper [9]. Overall, explanatory interaction can improve the model's quality.

Conclusion

We introduced explanatory interactive learning and proposed CAIPI, the first explanatory interactive learning method. CAIPI faithfully explains its queries in an interpretable manner and accounts for the user's corrections of the model if it is right (wrong) for the wrong the reasons. This opens the black-box of active learning and turns it into a cooperative learning process between the machine and the user. Our experimental results demonstrate that this cooperation can improve performance and indeed encourage (or discourages, if appropriate) trust into the model.

There are a number of interesting avenues for future work. Other interactive learning approaches such as coactive (Shivaswamy and others 2015), active imitation (Judah and others 2012), mixed-initiative interactive (Cakmak and others 2011) and guided probabilistic learning (Odom and Natarajan 2018) should be made explanatory. Making deep active learning (Gal and others 2017) explanatory is likely to improve upon the sample complexity of deep learning. Selecting queries that maximize the information of explanations, e.g., by using SP-LIME (Ribeiro and others 2016), as well as feeding back informative counterexample only are likely to improve performance.

Acknowledgments. The authors thank the anonymous reviewers as well as Antonio Vergari, Andrea Passerini, Samuel Kolb, Jessa Bekker, Xiaoting Shao, and Paolo Morettin for very useful feedback. ST acknowledges the supported by the European Research Council (ERC) under the European Unions Horizon 2020 research and innovation programme, grant agreement No. [694980] "SYNTH: Synthesising Inductive Data Models", and KK the support by the German Science Foundation project "CAML: Argumentative Machine Learning" (KE1686/3-1) as part of the SPP 1999 (RATIO).

[9] arxiv.org/abs/1805.08578

References

Attenberg, J., et al. 2010. A unified approach to active dual supervision for labeling features and examples. In *Proc. of ECML/PKDD*, 40–55. Springer.

Balcan, M.-F., et al. 2010. The true sample complexity of active learning. *Machine learning* 80(2):111–139.

Bastani, O., et al. 2017. Interpreting blackbox models via model extraction. *arXiv preprint arXiv:1705.08504*.

Bucilu, C., et al. 2006. Model compression. In *Proc. of KDD*, 535–541. ACM.

Cakmak, M., et al. 2011. Mixed-initiative active learning. *ICML 2011 Workshop on Combining Learning Strategies to Reduce Label Cost*.

Castro, R. M., et al. 2006. Upper and lower error bounds for active learning. In *The 44th Annual Allerton Conference on Communication, Control and Computing*, volume 2, 1.

Chen, L., et al. 2012. Critiquing-based recommenders: survey and emerging trends. *User Modeling and User-Adapted Interaction* 22(1-2):125–150.

Cortes, C., et al. 1995. Support-vector networks. *Machine learning* 20(3):273–297.

DeJong, G., et al. 2011. Explanation-based learning. In *Encyclopedia of Machine Learning*. Springer. 388–392.

Druck, G., et al. 2008. Learning from labeled features using generalized expectation criteria. In *Proc. of SIGIR*, 595–602.

Druck, G., et al. 2009. Active learning by labeling features. In *Proc. of EMNLP*, 81–90.

Gal, Y., et al. 2017. Deep bayesian active learning with image data. In *Proc. of ICML*, 1183–1192.

Hanneke, S., et al. 2014. Theory of disagreement-based active learning. *Foundations and Trends® in Machine Learning* 7(2-3):131–309.

Hoffman, R. R., et al. 2013. Trust in automation. *IEEE Intelligent Systems* 28(1):84–88.

Judah, K., et al. 2012. Active imitation learning via reduction to iid active learning. In *UAI*, 428–437.

Kimmig, A., et al. 2007. Probabilistic explanation based learning. In *Proc. of ECML*, 176–187.

Krause, A., et al. 2007. Nonmyopic active learning of gaussian processes: an exploration-exploitation approach. In *ICML*, 449–456. ACM.

Kulesza, T., et al. 2015. Principles of explanatory debugging to personalize interactive machine learning. In *Proc. of IUI*, 126–137.

Lundberg, S., et al. 2016. An unexpected unity among methods for interpreting model predictions. *arXiv preprint arXiv:1611.07478*.

Mitchell, T. M., et al. 1986. Explanation-based generalization: A unifying view. *Machine learning* 1(1):47–80.

Odom, P., and Natarajan, S. 2018. Human-guided learning for probabilistic logic models. *Front. Robotics and AI* 2018.

Peintner, B., et al. 2008. Preferences in interactive systems: Technical challenges and case studies. *AI Magazine* 29(4):13.

Raghavan, H., et al. 2006. Active learning with feedback on features and instances. *JMLR* 7(Aug):1655–1686.

Raghavan, H., et al. 2007. An interactive algorithm for asking and incorporating feature feedback into support vector machines. In *Proc. of SIGIR*, 79–86.

Ribeiro, M. T., et al. 2016. Why should i trust you?: Explaining the predictions of any classifier. In *Proc. of KDD*, 1135–1144.

Ribeiro, M. T., et al. 2018. Anchors: High-precision model-agnostic explanations. In *Proc. of AAAI*.

Ross, A. S., et al. 2017. Right for the right reasons: training differentiable models by constraining their explanations. In *IJCAI*, 2662–2670. AAAI Press.

Roy, N., et al. 2001. Toward optimal active learning through monte carlo estimation of error reduction. *ICML* 441–448.

Schnabel, T., et al. 2018. Short-term satisfaction and long-term coverage: Understanding how users tolerate algorithmic exploration. In *Proc. of WSDM*, 513–521. ACM.

Settles, B. 2011. Closing the loop: Fast, interactive semi-supervised annotation with queries on features and instances. In *Proc. EMNLP*, 1467–1478.

Settles, B. 2012. Active learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning* 6(1):1–114.

Sharma, M., et al. 2015. Active learning with rationales for text classification. In *NAACL HLT*, 441–451.

Shivaswamy, P., et al. 2015. Coactive learning. *J. Artif. Intell. Res.(JAIR)* 53:1–40.

Simpson, J. A. 2007. Psychological foundations of trust. *Current directions in psychological science* 16(5):264–268.

Small, K., et al. 2011. The constrained weight space svm: learning with ranked features. In *ICML*, 865–872. Omnipress.

Stumpf, S., et al. 2009. Interacting meaningfully with machine learning systems: Three experiments. *International Journal of Human-Computer Studies* 67(8):639–662.

Tong, S., and Koller, D. 2001. Support vector machine active learning with applications to text classification. *JMLR* 2(Nov):45–66.

Wang, N., et al. 2016. Trust calibration within a human-robot team: Comparing automatically generated explanations. In *Proc. of HRI*, 109–116.

Waytz, A., et al. 2014. The mind in the machine: Anthropomorphism increases trust in an autonomous vehicle. *Journal of Experimental Social Psychology* 52:113–117.

Zaidan, O., et al. 2007. Using "annotator rationales" to improve machine learning for text categorization. In *NAACL HLT*, 260–267.

Zaidan, O. F., et al. 2008. Modeling annotators: A generative approach to learning from annotator rationales. In *Proc. EMNLP*, 31–40.

Zhu, J., et al. 2004. 1-norm support vector machines. In *Proc. of NIPS*, 49–56.