Shapley Chains: Extending Shapley Values to Classifier Chains

Célia Wafa Ayad[1,2], Thomas Bonnier[2], Benjamin Bosch[2], and Jesse Read[1]

[1] LIX, École Polytechnique, Institut Polytechnique de Paris
[2] Société Générale

Abstract. In spite of increased attention on explainable machine learning models, explaining multi-output predictions has not yet been extensively addressed. Methods that use Shapley values to attribute feature contributions to the decision making are one of the most popular approaches to explain local individual and global predictions. By considering each output separately in multi-output tasks, these methods fail to provide complete feature explanations. We propose Shapley Chains to overcome this issue by including label interdependencies in the explanation design process. Shapley Chains assign Shapley values as feature importance scores in multi-output classification using classifier chains, by separating the direct and indirect influence of these feature scores. Compared to existing methods, this approach allows to attribute a more complete feature contribution to the predictions of multi-output classification tasks. We provide a mechanism to distribute the hidden contributions of the outputs with respect to a given chaining order of these outputs. Moreover, we show how our approach can reveal indirect feature contributions missed by existing approaches. Shapley Chains help to emphasize the real learning factors in multi-output applications and allows a better understanding of the flow of information through output interdependencies in synthetic and real-world datasets.

Keywords: Machine Learning Explainability · Classifier Chains · Multi-Output Classification · Shapley Values.

1 Introduction

A multi-output model predicts several outputs from one input. This is an important learning problem for decision-making involving multiple factors and complex criteria in the real-world scenarios, such as in healthcare, the prediction of multiple diseases for individual patients. Classifier chains [8] is one such approach for multi-output classification, taking output dependencies into account by connecting individual base classifiers, one for each output. The order of output nodes and the choice of the base classifiers are two parameters yielding different predictions thus different explanations for the given classifier chain.

To address the lack of transparency in existing machine learning models, solutions such as SHAP [5], LIME [9], DEEPLIFT [11] and Integrated Gradients [12]

have been proposed. Using Shapley values [10] is one approach to attribute feature importance in machine learning. The framework SHAP [5] provides Shapely values used to explain model predictions, by computing feature marginal contributions to all subsets of features. This theoretically well founded approach provides instance-level explanations and a global interpretation of model predictions by combining these local (instance-level) explanations.

However, these methods are not suitable for multi-output configurations, especially when these outputs are interdependent. In addition, the SHAP framework provides separate feature importance scores only for independent multi-output classifiers. By assuming the independence of outputs, one ignores the indirect connections between features and outputs, which leads to assigning incomplete feature contributions, thus an inaccurate explanation of the predictions.

Fig. 1 is a graphical representation of a classifier chain: patients with two conditions, obesity (Y_{OB}) and psoriasis (Y_{PSO}), given four features: genetic components (X_{GC}), environmental factors (X_{EF}), physical activity (X_{PA}) and eating habits (X_{EH}). From a clinical point of view, all factors X are associated with both conditions Y, obesity and psoriasis. However, since obesity is a strong feature for predicting psoriasis [4] (indeed, a motivating factor for using such a model is that predictive accuracy can be improved by incorporating outputs as features), it may mask the effects of other features. Namely, X_{PA} and X_{EH} will be found by methods as SHAP applied to each output separately to have zero contribution towards predicting Y_{PSO}, and one might interpret that psoriasis is mainly affected by factors which cannot be modified by the patient (environment and genetics). The *indirect* effects (physical activity and eating habits) will not be detected or explained.

We propose Shapley Chains to address this limitation of incomplete attribution of feature importance in multi-output classification tasks by taking into account the relationships between outputs and distributing their importance among the features with respect to a given order of these outputs. Calculating the Shapley values of outputs helps to better understand the importance of the chaining that connects these outputs and to visualize this relationship impact on the prediction of subsequent outputs in the chain. For these subsequent outputs, the computation of the Shapley values of the associated outputs shows the indirect influence of some features through the chain, which is generally not intuitive and missed by existing work. Our method will successfully explain these *indirect* effects. By attributing importance to the features X_{PA} and X_{EH}, Shapley Chains will help doctors to emphasize the importance of eating healthy and practicing physical activities in order to prevent and better cure psoriasis instead of blaming only genetics and exterior environmental factors.

This paper addresses the problem of attributing feature contributions in multi-output classification tasks with classifier chains when outputs are interdependent. Our contribution in this paper is resumed to :

– We propose Shapley Chains, a novel post-hoc model agnostic explainability method designed for multi-output classification task using classifier chains.

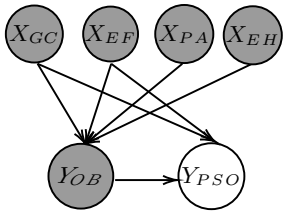


Fig. 1. An example of a multi-output task: predicting Y-outputs from X-features. A classifier chain uses the first output Y_{OB} as an additional feature to predict the second output Y_{PSO}.

- Shapley Chains attribute feature importance to all features that directly or indirectly contribute to the prediction of a given output, by tracking all the related outputs in the given chain order.
- Compared to existing methods, we show a more complete distribution of feature importance scores in multi-output synthetic and real-world datasets.

We devote Section 2 to a background and related work. In Section 3, we detail our proposed method Shapley Chains. Finally in Section 4, we run experiments on a synthetic and real-world datasets. The results of our method compared to SHAP values applied to independent classifiers are then discussed.

2 Background and Related Work

In this section we review multi-output classification, output dependencies, classifier chains and Shapley values to serve as a background for the rest of this paper. The notation we used is summarized in the next table.

Table 1. Notation

Notation	Meaning
$\mathbf{x}$	a given instance vector
$\mathbf{y}$	a given output vector
x_i	the i^{th} feature of instance $\mathbf{x}$
y_j	the j^{th} output
X	the feature space of x_i
Y	the output space of y_j
n	the number of features for each instance $\mathbf{x}$
m	the number of outputs

2.1 Multi-output classification and output dependencies

A multi-output classifier H is a mapping function that for a given instance $\mathbf{x}$=$\{x_1, x_2, ..., x_n\}$, such that $\mathbf{x} \in X$, it learns a vector of base classifiers $\mathsf{H}(\mathbf{x}) = h_1(\mathbf{x}), h_2(\mathbf{x}), ..., h_m(\mathbf{x})$ and returns a vector of predicted values $\mathbf{y} = \{y_1, y_2, ..., y_m\}$, with $y_j \in \{0, 1\}$ and $\mathbf{y} \in Y$.

In real-world applications, outputs can be dependent or independent. Designing classifiers that incorporate these output dependencies makes it possible to better represent the relationships in the data (between outputs, therefore between features and outputs). There are two types of output dependencies wrt subsequent outputs; namely marginal independencies, $P(\mathbf{y}) = \prod_{j=1}^{m} P(y_j)$, and conditional output dependencies:

$$P(\mathbf{y}|\mathbf{x}) = \prod_{j=1}^{m} P(y_j|X, y_1, ..., y_{j-1}) \tag{1}$$

In this article, we focus on output conditional dependencies. The nature of the relationship between features and outputs and between outputs is not restricted to causality. Therefore, no prior knowledge of the causal graph is necessary. This specific subject is partially covered in Shapley Flow [13], which is designed for single-output tasks.

2.2 Classifier chains

A classifier chain is one multi-output method that learns m classifiers (one classifier for each output, also referred as base classifier). All the classifiers are linked in a chain. The chaining method passes output information between classifiers, allowing this method to take into account output dependencies [7] when learning a given output in the chaining.

This method is exactly an expression of Eq. 1 if expressed according to the chain rule of probability (i.e., Fig. 2 as a probabilistic graphical model representation). That is one reason why conditional dependencies are interesting in this context. However, a classifier chain is not faithful to a 'proper' inference procedure, and rather takes a greedy approach to inference, plugging in predictions as observations; and proceeds much as a forward pass across a neural network. This creates some ambiguity between how much effect is gained from probabilistic dependence (as a probabilistic graphical model would) and feature effect (as one encounters via the latent layers of deep learning). Although discussion has been ongoing e.g., [8,7], there is not yet a consistent understanding in practice of what role a prediction plays as a feature to another label. By propagating output contributions among the features, Shapley Chains help to clarify these prediction roles, and confirm which outputs are interdependent using the Shapley value described in the next section.

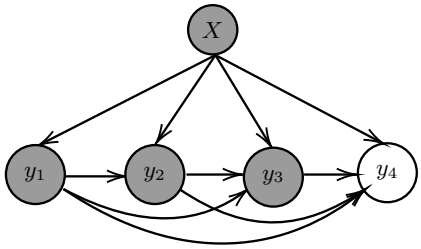


Fig. 2. One example of a classifier chain structure

2.3 Shapley values

The Shapley value expresses the contribution of feature x_i, to predict output y_j as a weighted sum:

$$\phi_{y_j} x_i \;=\; \sum_{S \subseteq X \setminus \{i\}} \frac{|S|!\;(|X| \;-\; |S| \;-1)!}{|X|!}\;[f_x\;(S \cup \{i\}) \;-\; f_x\;(S)] \qquad (2)$$

Where $S \subseteq X$, and f_x is the value function that defines each feature's contribution to each subset S. It computes each feature's average added value to each combination of features when making a prediction for instance $\mathbf{x}$.

Additivity is one axiom of a fair attribution mechanism that is satisfied by the Shapley value. It finds a good interpretation in multi-output classification. Consider two prediction tasks $(X,\ f)$, $(X,\ g)$ composed of the same set of features. We create a coalition prediction task $(X, f+g)$ by adding the two previous prediction tasks in the following way: $(f+g)(S) = f(S) + g(S)$ for all $S \subseteq X$. The additivity axiom states that the allocation of the prediction $(X, f+g)$ will be equal to the sum of the allocations of the two original prediction tasks. One should note that in this definition, we assume that the two prediction tasks are completely independent meaning that feature contributions to one prediction has no effect on the second one, which is not always the case because in real-world applications tasks are more often interdependent. One approach we propose is to use classifier chains because it permits to represent these relationships by introducing different chaining orders of these outputs. The overall feature Shapley values for a classifier chain can be calculated by marginalizing over all possible output chain structures. $\forall c \in \mathsf{C}$, the Shapley value of x_i in Eq. 2 can be written as follows:

$$\phi_{y_j x_i} = \frac{1}{|\mathsf{C}|} \sum_{c \subseteq \mathsf{C}} \phi_{y_j^c} x_i \qquad (3)$$

with $\phi_{y_j^c}$ being the contribution of feature x_i to the prediction of y_j with respect to the given chaining order c. For the matter of simplicity, we use ϕ_{y_j} to refer to $\phi_{y_j^c}$ in the rest of this paper. We report feature contribution for each chain

structure independently to show the impact of different chaining orders and the marginalization over these orders in Section 4.1.

2.4 Related work

The explainability of machine learning is an active research topic in the recent years. Several contributions have been made to explain single-output models and predictions. Inspecting feature importance scores of existing models is an intuitive approach that has served for many studies. These feature importance scores are either derived directly from feature weights in a linear regression for instance, or learned from feature permutations based on the decrease in model performance. Other more complex methods like LIME [9] learn a surrogate model locally (around a given instance) in order to explain the predictions of the initial model with simple and interpretable models like decision trees. On the other hand, DeepLift [11], Integrated gradient [12] and LRP [6] are some neural network specific methods proposed to explain deep neural networks.

The SHAP framework is one popular method attributing Shapley values as feature contributions. It provides a wide range of model-specific and model-agnostic explainers. Researchers have also proposed other Shapley value inspired methods incorporating feature interactions in the explanation process. For example, asymmetric Shapley values [3] incorporates causal knowledge into model explanations. This method attributes importance scores to features that do not directly participate in the prediction process (confounders), but fails to capture all direct feature contribution. On the other hand, on manifold Shapley values [2] focus on better representing the out of coalition feature values but provides misleading interpretation of feature contributions. Wang et al. [13] have proposed Shapley Flow, providing both direct and indirect feature contributions when a causal graph is provided. Resuming feature interactions to causality and assuming the causal graph is provided and accurate are two downsides of this method. These methods significantly contributed to advancing the explainability of machine learning models but none of them have tackled multi-output problems, more specifically when outputs are interdependent. Shapley Chains address this limitation.

3 Proposed Method: Shapley Chains

In this section, we introduce our approach to compute direct and indirect feature Shapley values for a classifier chain model. Note that our proposed method is model-agnostic, meaning that our computations do not depend directly on the chosen base learner used by the classifier chain.

We want to compute feature contributions to the prediction of each output $y_j \in Y$ for each instance $\mathbf{x}$. For example, Fig. 3 shows the direct and indirect contributions of x_i to predict output y_4 given in Fig. 2. In the next two sections, we detail the computations of the Shapley value of each feature to predict each output. We refer to these Shapley values as direct and indirect feature contributions.

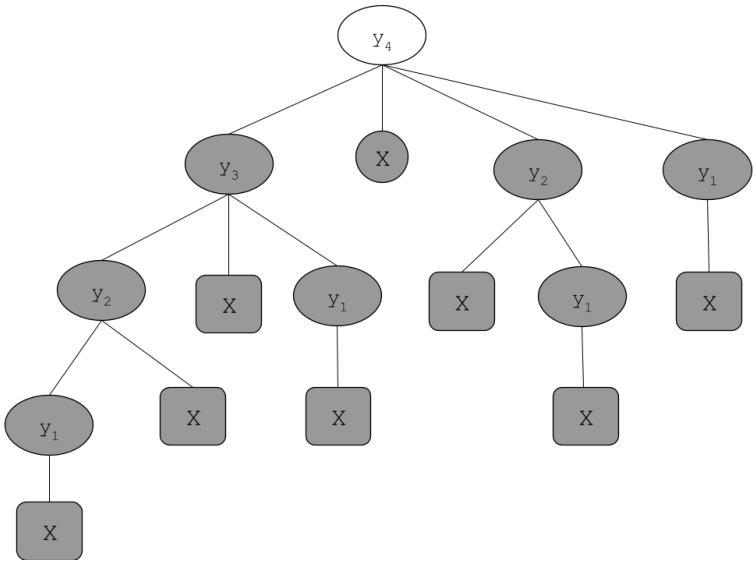


Fig. 3. Representation of direct and indirect contributions for a dataset with 4 outputs (y_1, y_2, y_3 and y_4). For example: the 4th output y_4 has 7 indirect Shapley values (7 paths ending with square leave) and one direct Shapley value (one path ending with a circle leaf).

Direct contributions The direct contributions are computed for features and outputs as in Eq. 2. Consider again the example of patients with the two conditions: psoriasis and obesity. For both Y_{OB} and Y_{PSO}, we use the framework SHAP in order to compute the Shapley value of each feature : X_{GC}, X_{EF}, X_{PA} and X_{EH}. This will attribute non zero Shapley values to X_{GC} and X_{EF} to predict Y_{OB} and Y_{PSO} separately. On the other hand, X_{EF} and X_{PA} will have non-zero Shapley values to predict Y_{OB} and zero values for the prediction of Y_{PSO}. The classifier chain method will add Y_{OB} to the feature set to predict Y_{PSO}. By running the SHAP framework on this new set, Y_{OB} will have a non zero Shapley value because it is dependent to Y_{PSO}. This Shapley value will be attributed to the features that are correlated to Y_{OB}. The attribution mechanism of direct feature (and output) contributions can be generalized to the classifier H with m base classifiers as shown in Algorithm 1.

For the first output y_1, we calculate the Shapley value of each feature according to Eq. 2, as done in the SHAP framework. This marginal value of all possible subsets to which the feature can be associated to is the feature's contribution to predict the first output y_1. For the second output y_2, we append the predictions y_1 made by the first classifier h_1 to the features set, and we train a second classifier h_2 to learn the second output y_2. We again use the SHAP framework to assign Shapley values to features and the first output y_1. Here, the feature set includes the first prediction. We perform the same steps for each

Algorithm 1 Computing direct feature contributions

procedure DICONTRIBUTION(X, Y, H) ▷ features, outputs, classifier chain model
  $i = j = 0$
  Φ=[]
  while $j < len(Y)$ **do**
    while $i < len(X)$ **do**
      $\Phi_{y_j} x_i \leftarrow SHAP(X, y_j, H)$ ▷ Shapley values of inputs wrt each output
      append y_j to X
      append $\Phi_{y_j} x_i$ to Φ
  return Φ ▷ Φ contains features and outputs Shapley values

remaining output. At each step, we calculate the Shapley values for features and previous predicted outputs that are linked via the chaining to the current output. At the final step, the feature set will contain n features and m outputs: $X = \{x_1, x_2, ..., x_n, y_1, y_2, ..., y_m\}$.

Indirect contributions The indirect contribution $\Phi_{indirect} y_j(x_i)$ of x_i to predict y_j is the weighted sum of the direct contributions of all $y_k \in Y$ that are chained to y_j. $\Phi_{indirect} y_j(x_i)$ is computed according to the Eq. 4.

$$\Phi_{indirect} y_j(x_i) = \sum_{k=1}^{j-1} \Phi y_j(y_k) \cdot Z_k(x_i) \tag{4}$$

where $j > 1$ and the function $Z_k(x_i)$ computes the weight vector for all paths from output y_k down to x_i. For $k > 1$ and $Z_1(x_i) = W(y_1, x_i)$, $Z_k(x_i)$ is recursively computed as follows:

$$Z_k(x_i) = \sum_{l=1}^{k-1} W(y_k, y_{k-l}) \cdot Z_{k-l}(x_i) + W(y_k, x_i) \tag{5}$$

where $W(y_k, y_{k-l})$ is the corresponding weight of y_{k-l} to predict the next output y_k (the direct contribution of y_{k-l} to predict y_k. And, $W(y_k, x_i)$ is the weight of x_i to predict y_k (the direct contribution of x_i to predict y_k). The weights $W(y_k, y_{k-l})$ and $W(y_k, x_i)$ are calculated according to:

$$W(y_k, .) = \frac{|\Phi y_k(.)|}{\left(\sum_{q=1}^{n} |\Phi y_k(x_q)| + \sum_{p<k} |\Phi y_k(y_p)|\right)} \tag{6}$$

where $\Phi y_k(x_q)$ is the direct contribution, as in Eq. 2, of each feature x_q to predict y_k). $p < k$ means the output p is chained to the output j forming a directed acyclic graph illustrated in Fig. 2.

For instance, in order to have a complete fair distribution of feature importance for the prediction of Y_{PSO}, we compute the indirect Shapley values of the features X_{PA} and X_{EH}. We do so by distributing the direct Shapley value of Y_{OB} computed previously to the four features. By the distribution operation, we

mean the multiplication of the direct Shapley value of each feature by the direct Shapley value of Y_{OB}, divided by the sum of the shapley values of all features for to predict the same output(here Y_{OB}).

We generalize this mechanism in Algorithm 2 of calculating indirect Shapley values to the chain structure in Fig. 2.3. The first output y_1 has always zero indirect Shapley values because there is no output that precedes it in the chaining. Thus, for the rest of this section, we compute feature indirect contributions for $y_j \in \{y_2, y_3, ..., y_m\}$. For each output y_j, there exists one direct path to the features thus one direct feature contributions and $2^j - 1$ indirect paths for each feature.

Algorithm 2 Computing feature indirect contributions

procedure INCONTRIBUTION(X, Y, Φ) ▷ inputs, outputs, Shapley values of features and outputs
  $i = j = 0$
  while $j < len(Y)$ **do**
    while $i < len(X)$ **do**
      compute $W(y_k, y_{k-l})$ and $W(y_k, x_i)$ in Eq. 6
      compute $Z_k(x_i)$ in Eq. 5
  return $\Phi_{indirect} y_j(x_i)$ in Eq. 4 ▷ returning indirect feature contributions.

One should notice that for the matter of the simplicity of understanding, we take the absolute value in Eq. 6. Thus, all the contributions will be positive. These absolute values can be replaced by the raw Shapley values in order to keep the positive or negative sign of feature contributions. Keeping the sign helps to understand if the feature penalizes or is in favor of the prediction.

4 Experiments

In order to assess the importance of the features that is attributed by our proposed framework[3] to explain their contributions to predict multiple outputs with a classifier chain, we run experiments on both synthetic and real-world datasets: a *xor* data that we describe next, and the Adult Income dataset from the UCI repository [1]. Here, we rely on human explanation to validate our results.

4.1 Synthetic data

To demonstrate our work, we first run experiments on a multi-output synthetic dataset containing two features (x_1 and x_2) and three outputs (*and*, *or* and *xor*) corresponding to the logical operations of the same names performed on x_1 and x_2. We split this dataset to 80% for the training and 20% for the test of our classifier.

[3] https://github.com/cwayad/shapleychains

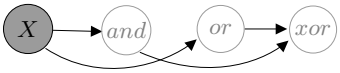


Fig. 4. The classifier chain structure for *xor* data. X is the set of features x_1 and x_2. *and*, *or* and *xor* are the outputs for which we want to compute direct and indirect Shapley values.

Next, we construct a classifier chain with the chaining order illustrated in Fig. 4. We use a logistic regression as the base learner. Our method is model agnostic meaning that it can be applied to a classifier chain with any other base learners. The use of the logistic regression as the base learner to predict *xor* is justified by the accuracy that this model achieves compared to other classifiers like decision trees. The classifier chain is trained on the train set using x_1 and x_2 to predict *and* and *or* separately. Then, we append these two predicted outputs to the features set in order to predict *xor*. Here, the order in which we predict *and* and *or* does not change our method's behavior.

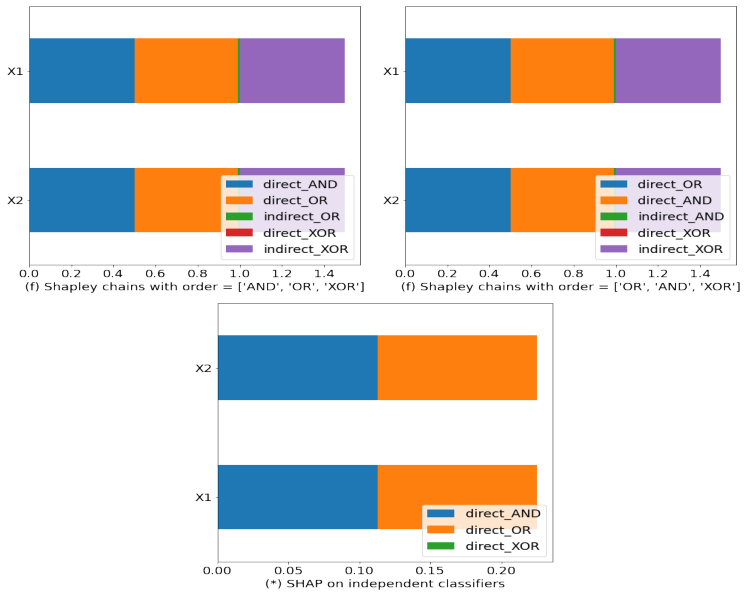


Fig. 5. A comparison of SHAP applied on independent classifiers and Shapley Chains. From the left to the right. (a) and (b) Normalized direct and indirect feature contributions made by Shapley Chains to predict *and*, *or* and *xor* for chain orders [*and*, *or*, *xor*] and [*or*, *and*, *xor*]. ($*$) SHAP assigns contributions to x_1 and x_2 only to predict *and* and *or* outputs and completely misses their contributions to predict *xor*. Absent colors refer to null Shapley values.

To explain the influence of x_1 and x_2 on the prediction of xor, we compared the application of the framework SHAP on each classifier independently and Shapley Chains on the trained classifier chain. We report our analysis on the test data. The results of the comparison shown in Fig. 5 indicate that the output chaining propagates the contributions of x_1 and x_2 to predict xor via and and or. Specifically, Fig. 5(a) and Fig. 5(b) illustrate that our method detects the indirect contributions of x_1 and x_2 (indirect_xor) to predict xor thanks to the chaining of and and or to xor implemented with the classifier chain model, which tracks down all feature contributions through the chaining of outputs. Furthermore, Fig. 5(a) and Fig. 5(b) confirm that predicting or before and or vice versa does not affect the feature contributions attribution, which confirms the chain structure for this data. On the other hand, these contributions of x_1 and x_2 are completely neglected by the SHAP framework on independent classifiers (Fig. 5($*$)).

Impact of the chaining order on the classifier chain explainability In order to measure the impact of the chaining order on the explainability of our classifier chain model with Shapley Chains, we performed analysis on the $3! = 6$ possible output chaining orders in the synthetic dataset (scenarios (a) and (b) in Fig. 5 and scenarios (c), (d), (e) and (f) in Fig. 6).

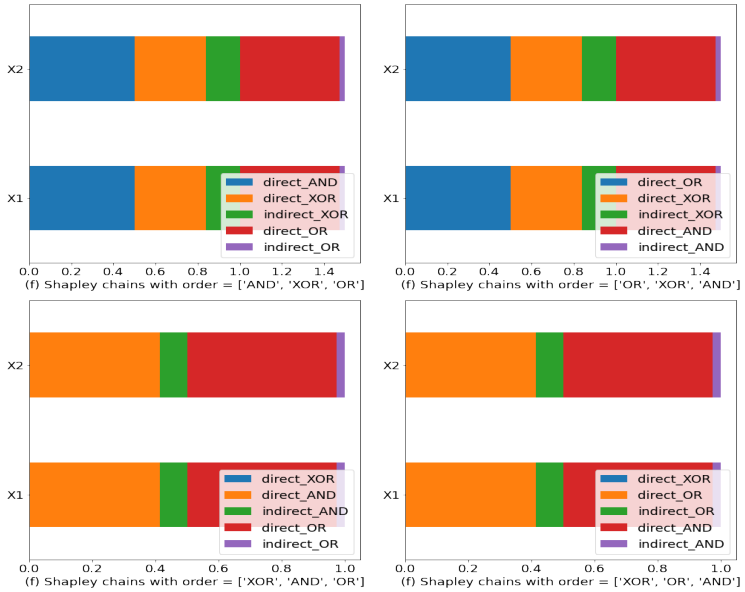


Fig. 6. Possible output chaining orders for xor data. Normalized total feature contributions (direct and indirect Shapley values) for c, d, e and f.

The information known to the classifier chain when training each output changes depending on the order of these outputs. For instance, in scenarios a and b (Fig. 5), we first learn the two outputs *and* and *or* using x_1 and x_2 features. *xor* is then predicted using *and* and *or*. Here, in both scenarios, both features x_1 and x_2 contribute indirectly (through *and* and *or*) to predict *xor*. Meanwhile in the scenario c (or d), the model relies on *and*(or *or*), x_1 and x_2 to predict *xor*. We observe that x_1 and x_2 have direct and indirect contributions, meaning that the classifier chain relies partially on these two features to predict *xor* (direct contributions of x_1 and x_2), and on *and* (indirect contributions of x_1 and x_2 via *and*). The last two scenarios e and f show no contribution of x_1 and x_2 to predict *xor*, which is explained by the fact that using only these two features, the model can not predict *xor* without having the information about the dependencies of *xor* to *and* and *or*.

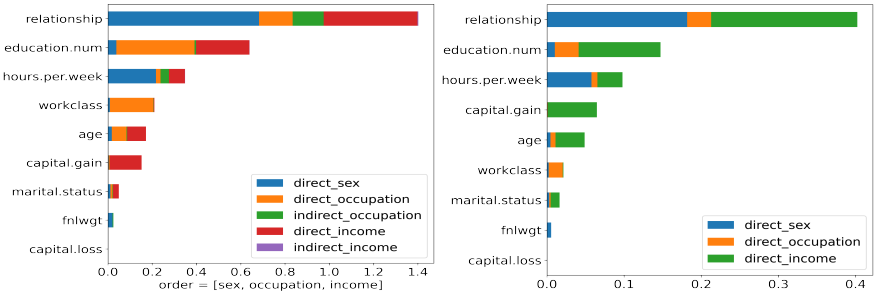
These results show that the chain order of *and*, *or* and *xor* outputs has an important role in the explainability of the classifier chain, because feeding different inputs to the classifier chain yields different predictions, thus different Shapley values are attributed to the features. x_1 and x_2 importance scores can either be derived from a direct inference of *xor* output only if there is additional information on output dependencies (for example *and* is linked to *xor*) or by extracting it from the chain that links *and* and *or* to *xor*. In the absence of all output dependencies of *and* or *or* to *xor*, the model completely ignores the importance of features x_1 and x_2 in the prediction of *xor*.

4.2 Explaining Adult Income with Shapley chains

We run Shapley Chains on the UCI Adult Income dataset. This dataset contains over 32500 instances with 15 features. We first discretize *workclass*, *marital status* and *relationship* characteristics. We remove *race*, *education* and *native country* and normalize the dataset with the min/max normalizer. Next, we split it into two subsets, using 80% for the training and the remaining 20% for testing. We evaluated the hamming loss of a classifier chain with different base learners and we kept the best base classifier, the logistic regression in this case.

In order to explain feature contributions to the predictions of the three outputs *sex*, *occupation* and *income*, we compared the results of Shapley Chains against classic Shapley values applied on separate logistic regression classifiers for different chain orders. Fig. 7 shows graphical representation of normalized and stacked feature contributions when applying Shapley Chains on our data set (Fig. 7 (a)), and stacked feature contributions from independent logistic regression classifiers (Fig. 7 (b)). In both cases, the magnitude of the feature contributions is greater in Shapley Chains compared to independent Shapley values, which confirms our initial hypothesis of some contributions are missed by SHAP framework, and these contributions can be detected when we take into account output dependencies. For example, the number of hours worked in a week (*hours.per.week*) has a more important indirect contribution to predict individual's *occupation* than a direct contribution. This is explained by the fact that *sex* is related to *occupation*, and this relationship is propagated

to the features by Shapley Chains. *relationship* is another example of Shapley Chains detecting indirect feature contributions to predict *occupation*. Furthermore, feature rankings are different in Shapley Chains. For example, the ranking of *capital.gain* comes in the fourth position (before *workclass*) using SHAP applied to independent classifiers. In our method, this feature's ranking is always less important (according to different chaining orders) than *workclass* to predict *sex*, *occupation* and *income* which makes more sens to us.



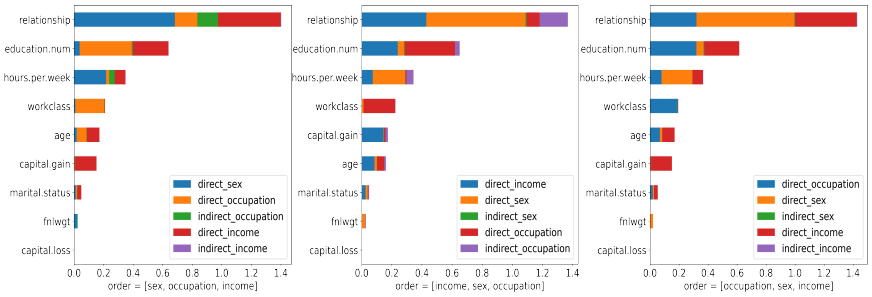
(a) Shapley Chains (b) SHAP on independent classifiers

Fig. 7. (a) Direct and indirect Shapley values on Adult Income data: we normalize and stack each feature's direct and indirect contributions to each output. *sex* has only direct contributions because it is the first output we predict in this chain order. (b) Stacked Shapley values of independent classifiers on Adult Income data.

We also tested the impact of different chain orders of these three outputs on the feature importance attribution. Fig. 8 illustrates three different chaining orders. Each different order allows each classifier to use different prior knowledge to learn these outputs. For example in Fig. 8(b), we first predict *income* and *sex* and we use this information to predict *occupation*. Intuitively, *occupation* is correlated to individual's *sex* and *income*. The classifier chain uses this information provided to the third classifier to predict *occupation*. Here, Shapley Chains attribute more importance to the factors that predict both *income* and *sex*, when predicting *occupation*. Shapley Chains preserve the order of feature importance scores across all the chaining orders in general, but the magnitude of each feature's importance differs from one chain to another. This is due to the prior knowledge that is fed into the classifier when learning each output. In addition, these feature importance scores are always more important in Shapley Chains compared to Shapley values of independent classifiers for all chain orders.

5 Conclusions and Perspectives

In this paper, we presented Shapley Chains, a novel method for calculating feature importance scores based on Shapley values for multi-output classification with a classifier chain. We defined direct and indirect contribution and



(a) (b) (c)

Fig. 8. Stacked direct and indirect feature effects for 3 different chain structures over Adult Income data.

demonstrated on synthetic and real-world data how the attribution of indirect feature contribution to the prediction is more complete with Shapley Chains. Our method helps practitioners to better understand hidden influence of the features on the outputs by detecting indirect feature contributions hidden in output dependencies. Although the rankings of feature importance are not always different from independent feature importance scores, the magnitude of these scores is always important in Shapley Chains, which is more important to look at in applications that are sensitive to the magnitude of these importance scores rather than their rankings. By extending the Shapley value to feature importance attribution of classifier chains, we make use of output interdependencies that is implemented in classifier chains in order to represent the real learning factors of a multi-output classification task.

To extend this work, Shapley Chains could be evaluated on multi-output regression tasks. Exploring the relationship's type between the outputs, and studying wether Shapley Chains preserve all these relationships when attributing feature contributions is another open question of our work.

References

1. Dua, D., Graff, C.: UCI machine learning repository (2017), http://archive.ics.uci.edu/ml
2. Frye, C., de Mijolla, D., Begley, T., Cowton, L., Stanley, M., Feige, I.: Shapley explainability on the data manifold (Dec 2021)
3. Frye, C., Rowat, C., Feige, I.: Asymmetric Shapley values: Incorporating causal knowledge into model-agnostic explainability (Dec 2021)
4. Jensen, P., Skov, L.: Psoriasis and Obesity. Dermatology (Basel, Switzerland) **232**(6), 633–639 (2016)
5. Lundberg, S., Lee, S.I.: A Unified Approach to Interpreting Model Predictions (Nov 2017)
6. Montavon, G., Binder, A., Lapuschkin, S., Samek, W., Müller, K.R.: Layer-Wise Relevance Propagation: An Overview. In: Samek, W., Montavon, G., Vedaldi, A., Hansen, L.K., Müller, K.R. (eds.) Explainable AI: Interpreting, Explaining and Visualizing Deep Learning, vol. 11700, pp. 193–209. Springer International Publishing, Cham (2019)
7. Read, J., Pfahringer, B., Holmes, G., Frank, E.: Classifier chains for multi-label classification. Machine Learning **85**(3), 333–359 (Dec 2011)
8. Read, J., Pfahringer, B., Holmes, G., Frank, E.: Classifier Chains: A Review and Perspectives. Journal of Artificial Intelligence Research **70**, 683–718 (Feb 2021)
9. Ribeiro, M.T., Singh, S., Guestrin, C.: "Why Should I Trust You?": Explaining the Predictions of Any Classifier (Aug 2016)
10. Rozemberczki, B., Sarkar, R.: The Shapley Value of Classifiers in Ensemble Games (Jun 2021)
11. Shrikumar, A., Greenside, P., Kundaje, A.: Learning Important Features Through Propagating Activation Differences (Oct 2019)
12. Sundararajan, M., Taly, A., Yan, Q.: Axiomatic Attribution for Deep Networks (Jun 2017)
13. Wang, J., Wiens, J., Lundberg, S.: Shapley Flow: A Graph-based Approach to Interpreting Model Predictions. In: Proceedings of The 24th International Conference on Artificial Intelligence and Statistics. pp. 721–729. PMLR (Mar 2021)