Path-Based Visual Explanation

Mohsen Pourvali(✉), Yucheng Jin, Chen Sheng, Yao Meng, Lei Wang,
Masha Gorkovenko, and Changjian Hu

Lenovo, Beijing, China
{mpourvali,jinyc2,shengchen1,mengyao1,wanglei63,mgorkovenko,
hucj1}@lenovo.com

Abstract. The ability to explain the behavior of a Machine Learning (ML) model as a black box to people is becoming essential due to wide usage of ML applications in critical areas ranging from medicine to commerce. Case-Based Reasoning (CBR) received a special interest among other methods of providing explanations for model decisions due to the fact that it can easily be paired with a black box and then can propose a post-hoc explanation framework. In this paper, we propose a CBR-Based method to not only explain a model decision but also provide recommendations to the user in an easily understandable visual interface. Our evaluation of the method in a user study shows interesting results.

Keywords: Explainable artificial intelligence · Case-based reasoning · Classification

1 Introduction

The ability to explain the behavior of a ML model to people is becoming essential due to wide usage of ML applications in critical areas ranging from medicine to commerce.

Most of the current explanation methods assign scores to *input* features, by which features that have highest influence on the model's decisions are identified. Explaining the underlying reasons for an image classification model's decision to a human is easier than explaining the decision of a text classification model.

In an image, we can represent segments of the image as concepts [3], and a model's decision can be understood by a human if we explain it using these concepts. For example, in an image of a girl, her hair is a concept, and it is easily understandable for an AI novice if we explain to her that this part of the image was the main reason that model thinks this is a picture of a girl. The understandable concepts in text could be a sentence or paragraph, like saying which sentence is the main reason for a model decision. But, in tabular data, it is hard to explain the reasons for a model's decision through concepts. We define this difficulty as **Feature Inability** and **Feature Ambiguity** problems.

X. Zhu et al. (Eds.): NLPCC 2020, LNAI 12431, pp. 454–466, 2020.
https://doi.org/10.1007/978-3-030-60457-8_37

Feature Inability: The first problem is that each feature in tabular data individually (like a single pixel in an image) is not able to explain the reason behind the model's decision, and also unlike what a collection of pixels in images as segments/super-pixel of image can do for understandability of an explanation, a collection of features in tabular data is not quite understandable by a human since finding the relations between the features is difficult.

Feature Ambiguity: Another problem is that the meaning of a feature solely in tabular data might be ambiguous to a human, like a single pixel in an image is. But, unlike images, in which a collection of coherent pixels makes it understandable for a human, in tabular data even a collection of features still doesn't change the ambiguity of the features. For instance, in an image of a dog, a single pixel is not meaningful, but it is possible to select a segment of pixels (e.g., the dog's leg) which is understandable for a human. In tabular data, features are like scattered pixels in that even a collection of them are possibly not understandable to a human.

A collection of coherent pixels as a segment of an image shows relations between the features/pixels in that segment which are understandable and meaningful to human eyes. But in tabular data, even if there is a relation between two features, it might be unclear to a human. And this is because of the nature of images: images come from a real world object, so all features already are defined and are put together, and there is an order that makes it easy to separate features in meaningful segments. Also, in text data, we can see this meaningful order and segmentation (e.g., sentence, paragraph) but not exactly like what exists in images. In tabular data, there is no meaningful order and segmentation in features. The two above mentioned issues also indicate the importance of a visualization of the explanation that enables user to interpret the underlying reasons for a model decision. A good explanation for a model's prediction of an instance may not be a complete explanation of all the reasons for the prediction, but it could be a contrastive explanation comparing how the prediction is different from the prediction for another instance [11]. Another factor for a good explanation could be providing a few recommendations to the user. For example, if you apply for a loan and your application is rejected, you might want to not only know the reasons but also to understand the agent's reasoning in a bid to strengthen your next application [9].

We propose a Hybrid method to address these issues for any probability-based classification model. We first establish an explanation algorithm by taking advantage of *Facts* and *Foils* concepts. Regarding explanations, people are not only interested in why event P happened, they also want to know why not event Q happened instead of event P. The event that did happen P is referred to as Fact, and the contrasting event that did not happen Q is referred to as Foil [8]. A Foil could be any sample to be compared with a Fact. We consider the better samples and the best sample in a Path-Based fashion to compare with a Fact. This explanation exposes more hidden knowledge of the samples, model, and data to users. Indeed, we try to answer three questions which may also be asked by the user, which are: why did a *better* event not happen; why did the *best* event not happen; and an important question, which is how to touch these events. We then present our Path-Based explanation with a visual interface which is easily understandable for a user.

2 Related Works

CBR enables us to present a post-hoc mechanism to not only predict the model result of a query case, but also explain the model's decision by using examples which are similar to the case with respect to the model. Indeed, CBR as a more interpretable system can be paired with a black box in a way that provides explanatory samples based on model prediction, to generate a *twin-system* [5]. Authors in [5] survey similar approaches that use CBR in a post-hoc fashion as one particular solution to the eXplainable Artificial Intelligence (XAI) problem. For example, [1] uses a learned model as distance metric to find explanatory cases for an Artificial Neural Network (ANN) in the medical domain. Authors use Euclidean distance to measure similarity between latent features (i.e., hidden units activation vector of ANN model) of the case to be explained and all the training dataset, and then they present cases with small Euclidean distance to the query case as the similar cases to the query case to then explain the ANN reasoning for the query case. In [12], authors select explanatory cases based on their similarity in their local important features to the query case to be explained. [2] evaluate the usefulness of CBR in terms of retrieving explanatory cases to explain a prediction, and show that it is more convincing than an explanation based on rules. Visualization of CBR-paired systems can even enhance transparency and understandability of the proposed explanation. [10] show that knowledge-intensive tasks require a better explanation than just a set of retried cases. Local information of a query case that enables the user to easily identify similarity of the cases must be visible to the user. [7] proposes a CBR system able to classify a query case using an automatic algorithm, but also through visual reasoning. Authors in [7] select similar cases from the feature/input space of the model.

This work is inspired by [7], our approach in this context is a post-hoc approach that explains the underlying reasons for a model decision, in which similar points are selected in the model result/output space. In our approach the samples sit next to each other for a specific goal, which is to build a path from the query case to the best case in each class. These samples are selected from the candidate cases that their model results are close to, and a direct line in the model output space is drawn between the query case result and the best case. We only use three colors in the visual interface, which makes it easier for the user to identify the dominant color. It also can be modified for colorblind people by using different shapes for each color. Providing a path from the model output of a query case to the best result of the model can depict an **evolution** process, and in turn can help the user to understand how (s)he can get a better result from the model. Furthermore, it can provide **recommendations** for this aim.

3 Our Proposed Method

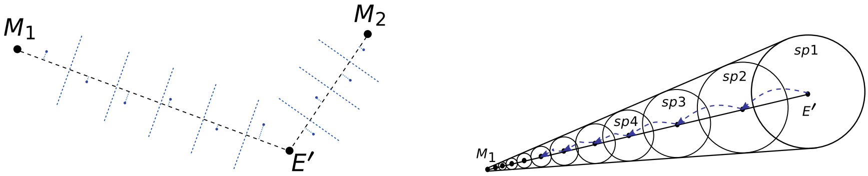
Providing a path from a model's result of a sample case to the best result of the model can depict an evolutionary process, and in turn can help the user to understand how (s)he can get a better result from the model. Furthermore, a path on which there are several cases from other classes implies a form of analogical reasoning: Case-Based Reasoning (CBR) in which the solution for a new case is determined using a database of previous known cases with their solutions. Cases similar to the new case are retrieved from the database, and then their solutions are adapted to the case. This situation provides an Interpretable Classification, in which a user can classify the new case according to his own knowledge and the knowledge retrieved from CBR. In this paper we do not use CBR to classify new cases; we select similar/explanatory cases from the *output* of a ML model to visually explain the model's result for a query case. The proposed visual interface aims at identifying what is the dominant color? However, this explanation can either be a supporting or nonsupporting proof for the model decision.

A path from a query case to the best case of each class in the model result space provides a better understanding of the model due to evaluation of the similar cases that appear on the path. A visual interactive explanation with an embedded path that constructs CBR in ML results space provides a transparent insight of the ML, which can be used also to evaluate different ML models. Each specific model has its own best case, path, and explanatory cases on the path.

Assuming a classification model with N classes $\mathbb{C} = \{C_1, C_2, .., C_N\}$ trained on a training dataset and testing dataset, for a single case E as input to be explained and its probability E' as result of the model for that case, our explanation algorithm works as follows: It selects two classes as default, the class of the query case result and the class with the highest probability (the selection can also be based on the user desire), and then it generates two paths, each one from E' to a point M_i which is the best result (i.e., highest probability) obtained for a case in class C_i. A path is generated by connecting a collection of points in the probability space that are very close to the direct line connecting E' to M_i in 3-Dimension space.

In general, the workflow of the approach has two steps. In the first step, a 3-Dim model results space is generated and two paths with similar cases in each are indicated. In the second step, a visual explanation for input/query case is presented.

First Step: Given a vector $E = [e_0, e_1, e_2, ..., e_{N-1}]$ with N dimensions as a distribution over N classes for a classification model with an input case p_i, (a) two classes C_1 and C_2 based on user desire or default classes are selected, and the rest of the classes' probabilities are reduced to one dimension e'_3 to generate a new vector $E' = <e_1, e_2, e'_3>$. We use Multidimensional Scaling (MDS) for dimension reduction to preserve distances involving the query case, (b) from each class, the best sample is selected which is a result of the model for a sample



(a) Two paths to the best cases in class C_1 and class C_2

(b) Various distributions of Explanatory Samples over the path

Fig. 1. 2-dim visual illustration of paths and explanatory samples space.

that has highest probability in corresponding dimension of its distribution. (c) two paths from E' to C_1 and C_2 are conducted by identifying the nearest cases to the path as Explanatory Cases (EC), shown in Fig. 1a. In order to depict the evolution process for the sample case E', each path is divided into several areas, and from each area an EC is selected. Indeed, these ECs build the paths through which we can see how features of a sample case are changed to reach the best result in each class. Each EC is a case from the corresponding class for which the result of the model is close to the direct line/path between E' and the best result in that class. Indeed, the path is a direct line in 3-Dim between the model result of the query case and the best case, and ECs are the closest point in model result space to this line. The Explanatory Cases are selected from the testing data, which is a small portion of ground truth cases. This reduces the computational/memory allocation cost (specially MDS cost), and it is able to provide a comparable environment by using different and new testing data that introduces new best cases, various paths and recommendations, which in turn provides comparison metrics to evaluate different ML models. We select ECs from inside different step areas separated with dotted lines perpendicular to the paths shown in Fig. 1b. These areas are not necessarily equal areas, since the distribution of ECs over an specific path is not normal, thus, the more dense the distribution, the more ECs are selected. For example, assume that the distribution of points close to the query case is dense(bigger cycles in Fig. 1b) and density is being reduced by getting away from the query case. In this case, more ECs are selected from the area around the query case. To implement this, we first map all of the candidate ECs (i.e., close to the path) to a one-dim array, and then by using a constant distance of index in the array we select one EC from each area.

The distance between a point (model result) and direct line (path) is calculated using the following formula:

$$d_i = \frac{\left|\overline{p_i E'} \times \overline{s}\right|}{|\overline{s}|} \tag{1}$$

Algorithm 1. Algorithm for recording the recommendations

```
function SuggRecord($E, EC_i$)
    for all not equal $f$ in Features($E, EC_i$) do
        $Combs = Coalitions(f, coalition\_length)$
    for $c$ in $Combs$ do
        Swap($E, EC_i, c$)
        $dis_1 = 1$ - OneDimDistance($E', p_i$)
        $dis_3 = 1$ - Euclidean($E', p_i$)
        $min = 1$ - $(\alpha \times dis_1 + (1 - \alpha) \times dis_3)$
        Record $f$ with lowest $min$
```

where $p_i = [p_{i0}, p_{i1}, p_{i2}]$ is the probability vector of a candidate explanatory case EC_i, E' is the query case, and $s = [(e'_0 - p_{i0}), (e'_1 - p_{i1}), (e'_2 - p_{i2})]$ is the directing vector of the line.

We use a weighted linear combination of Euclidean and one-dimensional distance to record the recommendations for each pair of the query case and explanatory case, shown in Algorithm 1. The goal is to minimize the distance between the model result of the query case and a specific explanatory sample on the path. Indeed, similar to Shapley Values [14], we try to find a feature's value that has the highest contribution to increase probability of the query case in a class. But here, there is only one sample, i.e., the sample in the step that we want to get there, and coalitions of features are limited to those which are not equal in value, comparing features of E and EC_i[1].

Second Step: In the second step, we generate a Visual Explanation as shown in Fig. 2, which is inspired by Rainbow Boxes [6]. As it is shown in Fig. 2, the corresponding model's input case query for vector E' is in the middle of the explanation, and the best case for each class is located at each corner. The two classes at corners of the explanation are based on user desire or are default classes. Characteristics of the visual interface are explained in Sect. 4.

Our proposed explanation for a single case in tabular data can address the two problems mentioned before. Regarding Feature Inability, using CBR with cases which are considered to be better than the query case provides an understandable explanation for the user, by allowing comparison of a collection of connected features that have the same path in common. Regarding Feature Ambiguity, a path that explains the evolutionary process of changing a feature's value to enhance the probability of being selected as a better member of a class (in model point of view), and building a coalition of cases with similar or different feature's value all aims at one goal: helping to disambiguate the features and their relations.

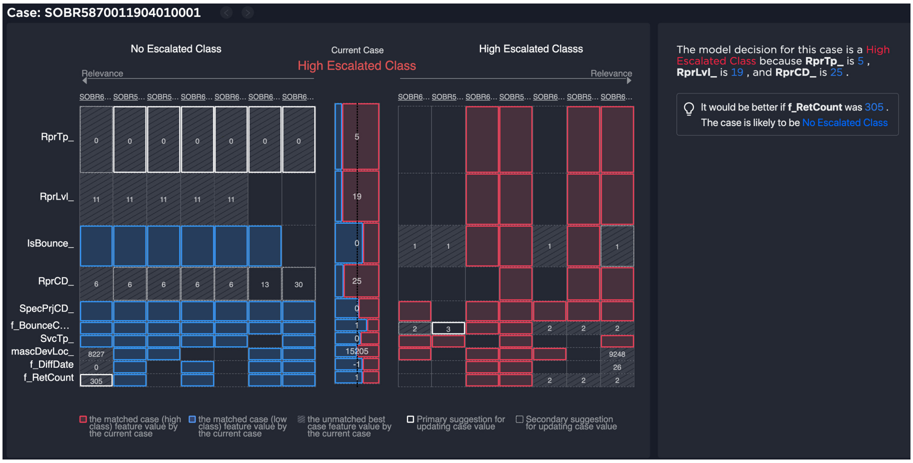
[1] In our experiments we use coalitions with only a single member.

4 Visual Interface

Figure 2 shows the visual interface designed for an ML classification model that predicts *legal* cases, and that its three target classes are *No*, i.e., the case is not legal, *Low*, i.e., the case is legal with low level, and *Hight*, i.e., the case is legal with high level. Two user-desired classes are *High* and *No* corresponding to C_1,C_2, respectively, and important local features identified by LIME are shown on the left side. The characteristics of the Visual Explanation are as follows:

- The ECs on each path are identified by different colors corresponding to different classes, e.g., Red for class High and Blue for class No, shown in Fig. 2.
- The value inside each box is the feature value; thus, the user is able to explore the feature's change through each path to the best result of the model.
- As it can be seen in Fig. 2, the length of each box is different, and it is proportional to the importance of the corresponding feature for that box. For example, the feature placed in the first row is the most important one. To rank importance, we use LIME [13] with the aim of finding a local feature importance for query case.
- Looking at the dominant color in Fig. 2, the user can recognize at a glance that class *High* is a better choice to classify the query case, and *High* is indeed the model target result for the query case.
- Furthermore, we can see more information in Fig. 2, like a suggestion which represents how we can get a better result from the model. For example, if we walk in the path to class *No*, we will get a better result for the query case if we only replace value of feature *RprTp_* by 0 instead of 5. In other words, if we want to have a better result of the model, and we can only go one step toward the best result of the model in class *No*, and also are allowed to only change one feature's value, then, feature *RprTp_* would be one of the best features and for which value 0 is one the best value to choose. We identify this information by replacing a feature's values of the query with the feature's values of the specific sample on the path, shown in Algorithm 1.
- At top right of the interface in Fig. 2, by applying natural language, the result of LIME is presented in understandable way for the user. We also use this result to examine visual-based and text-based explanations.
- Right below the LIME result there is a recommendation panel in 2, this part shows the first possible recommendation directing to the best sample in a target class. The first and second recommendations for each step in each direction are shown with thick and dotted outline borders for the corresponding feature box, respectively.
- Another piece of information that we can see in Fig. 2 is the priority of a feature's value. On the path to the best sample in class *No* it is shown that from sample 1 to sample 6 the feature whose value is best replaced with query case is *RprTp_*; but for the last sample, which is the best sample also, the best feature becomes *f_RetCount*. Indeed, for sample 6 and the best sample, all of the important features have the same values, and it is expected that

still the value of feature *RprTp_* will be the best choice to be replaced with 5. But as it is shown, *f_RetCount* is the first recommendation, since the value for *f_RetCount* in all the samples except the last one is less than 2. Considering the two last samples, which are the same in most important features, it shows that value 305 for feature *f_RetCount* has a higher impact in class *No* compared to the value *0* for feature *RprTp_*.


Case: SOBR5870011904010001

The model decision for this case is a High Escalated Class because **RprTp_** is 5 , **RprLvl_** is 19 , and **RprCD_** is 25 .

It would be better if **f_RetCount** was 305 . The case is likely to be No Escalated Class

No Escalated Class
Current Case
High Escalated Class
High Escalated Classs
Relevance
Relevance
SOBR6... SOBR5... SOBR5... SOBR6... SOBR6... SOBR6... SOBR6...
SOBR6... SOBR5... SOBR6... SOBR5... SOBR6... SOBR5... SOBR6...
RprTp_
RprLvl_
IsBounce_
RprCD_
SpecPrjCD_
f_BounceC...
SvcTp_
mascDevLoc_
f_DiffDate
f_RetCount
0 0 0 0 0 0 0
5
11 11 11 11 11
19
0
1 1 1 1
6 6 6 6 6 13 30
25
0
1
2 3 2 2 2
0
8227
15205
9248
0
-1
26
305
1
2 2 2
🟥 the matched case (high class) feature value by the current case
🟦 the matched case (low class) feature value by the current case
▨ the unmatched best case feature value by the current case
□ Primary suggestion for updating case value
□ Secondary suggestion for updating case value

Fig. 2. A snapshot of the visual interface

The core of the visual interface is written in the Python language. The application backend service uses the Java language to unify processes, and the frontend is uniformly built using VUE. Due to the large latency of python core processing data, an asynchronous interaction is established through Kafka as a message middleware.

5 Experimental Setup

To evaluate our designed visualization, we measure the user-perceived quality of the visualization by using the System Causability Scale [4], which is a simple and rapid evaluation tool to measure the quality of an explanation interface or an explanation process itself.

5.1 Dataset

We used an imbalanced dataset consisting of about 1 million real cases logged in a repair center for mobile devices. This data is used to train a classification model with 30 input features to classify escalation of a case into 3 classes *No*, *Low*, and *High*. The visualization shows how a queried case is likely to match two selected classes based on the case-based reasoning algorithm.

Table 1. Ten question items of System Causability Scale

Statements
1. I found that the data included all relevant known causal factors with sufficient precision and granularity.
2. I understood the explanations within the context of my work.
3. I could change the level of detail on demand.
4. I did not need support to understand the explanations.
5. I found the explanations helped me to understand causality.
6. I was able to use the explanations with my knowledge base.
7. I did not find inconsistencies between explanations.
8. I think that most people would learn to understand the explanations very quickly.
9. I did not need more references in the explanations: e.g., medical guidelines, regulations.
10. I received the explanations in a timely and efficient manner.

5.2 Evaluation Measures

We compose a questionnaire based on the System Causability Scale which consists of ten statements (Table 1). Participants are asked to rate each statement by using a five-point likert scale that ranges from strongly agree to strongly disagree. In the end, the quality of visualization is indicated by the average rating of ten statements $SCS = \sum_i rating_i/5 * 10$.

In addition, we also asked three additional questions to collect the subjective feedback for the visualization.

1. How do you think the visualization can help you make a decision?
2. Is it more likely that you trust the prediction result when the visualization is presented? Why?
3. Which one (visual explanation versus textual explanation) is more effective for increasing the transparency of reasoning algorithm?

5.3 Study Procedure

We asked participants to follow the following procedure to perform a task by using the presented visualization.

Task: Based on the visualization in Fig. 2, please reduce the risk of escalation for the queried case by adjusting its feature values. i.e. convert a case of high escalation to a case of no escalation. To better judge if participants understand the visualization, the task includes a restriction that the value of feature $RprTp\_$ is not allowed to change.

1. The participants were asked to attend a training to get familiar with the experimental task and the main functions of visualization.

2. After finishing the training, the participants write down how to adjust the feature value of queried case with the purpose of *no escalation*.
3. Finally, the participants filled out the questionnaire and answered three open questions.

5.4 Participants

We recruited 5 participants from a high-tech company to test the visualization based on a given task. The demographics of the participants are shown in Table 2.

Table 2. Participants' demographics.

ID	Age	Sex	Occupation	Working experience (years)	Education
1	42	Male	Visual designer	20	Bachelor
2	35	Female	Interaction designer	10	Master
3	28	Male	Software developer	6	Bachelor
4	28	Female	Visual designer	6	Bachelor
5	27	Female	Visual designer	5	Bachelor

6 Results and Discussions

6.1 Objective Results

We measure the actual quality of visualization by the effectiveness of the actions (Table 3) the participants took to reduce the escalation risk for the queried case. The result shows that three of the five participants took actions that were exactly the same as the ones that the system suggested. Although P1 did not take the optimal action, P1's actions are still reasonable for the task goal. P4's action seems to be not logical since the value of feature RprLvl_ is not 1 for all presented cases.

Table 3. Actions taken by the participants.

ID	Actions
1	RprLvl_: 19→11, RprCD_: 25→6
2	mascDevLoc: 15205→8227, f_RetCount: 1→305
3	RprLvl_: 19→1
4	RprLvl_: 19→11, f_RetCount: 1→305, RprCD_: 25→6
5	f_RetCount: 1→305, RprCD_: 25→6

6.2 Questionnaire Results

Figure 3 shows the results of participants responses on the System Causability Scale (SCS). The average score of SCS is 0.588 and standard deviation is 0.143. Although the score does not indicate a good quality of explanation according to the reference value 0.680, the visualization still is rated high for some aspects such as 5. Understanding causality, 7. No inconsistencies, and 10. Efficient.

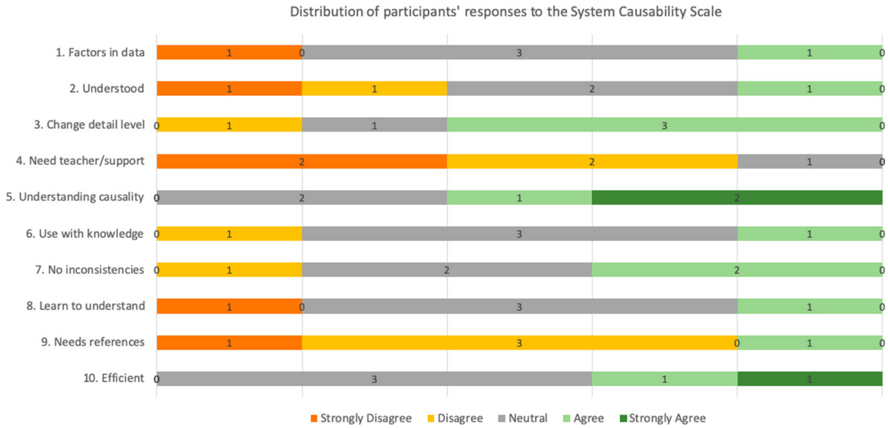


Fig. 3. Distribution of participants' responses to the System Causability Scale

In addition, all participants think that the visualization can help them make a decision if they have been trained for using this visualization. As we assumed, all participants state that they tend to trust the prediction result more if the visualization is presented. However, regarding a preferred method of explaining the case-based reasoning, not all participants prefer the visualization because they can simply know how to achieve their goal just by following the textual suggestion, and the complexity may also hinder them from using the visualization properly. E.g., they were struggling with understanding the way that the weight of features was presented and the relevance of the case in each escalation class. The participants who are in favor of visualization thought it allows them to freely explore the system and deeply understand the logic of reasoning.

6.3 Discussion

Overall, despite the high complexity, most of the participants value the visualization in terms of **understanding causality, efficiency, support in decision making, and user trust**. After a simple training, four out of five participants can take an optimal action to decrease the escalation class without violating the restriction of adjusting feature, which implies that participants are able to trade off among multiple features that can be adjusted.

The overall score of SCS is lower than the suggested score for good quality of explanation. Arguably, this visualization is designed for users with professional knowledge in a specific application domain. However, all participants do not have knowledge on repairing service of mobile phones needed in the user scenario. Therefore, most participants reported that they need substantial support and training before using the visualization.

7 Conclusion

We proposed a visual explanation based on an evolutionary path through CBR. We discussed the difficulty of explaining model decisions in tabular data, and inability and ambiguity of single features in this data. We then presented a coherent visual explanation by which a user can see the relation between samples and features through a set of connected samples, which are placed side by side each other with one step improvement in quality between them. Our experiments showed that, by answering the three questions implied by the Fact and Foil concept (why not a little better event, why not the best event, and how to achieve the event) a user can better understand a the decision of a model that uses tabular data.

In the future we intend to extend this work to other data types. We want to expand this method in text data by exploiting a knowledge graph to also visualize semantic relations of samples and features through an evolutionary path explanation.

References

1. Caruana, R., Kangarloo, H., Dionisio, J., Sinha, U., Johnson, D.: Case-based explanation of non-case-based learning methods. In: Proceedings of the AMIA Symposium, p. 212. American Medical Informatics Association (1999)
2. Cunningham, P., Doyle, D., Loughrey, J.: An evaluation of the usefulness of case-based explanation. In: Ashley, K.D., Bridge, D.G. (eds.) ICCBR 2003. LNCS (LNAI), vol. 2689, pp. 122–130. Springer, Heidelberg (2003). https://doi.org/10.1007/3-540-45006-8_12
3. Ghorbani, A., Wexler, J., Zou, J.Y., Kim, B.: Towards automatic concept-based explanations. In: Advances in Neural Information Processing Systems, pp. 9277–9286 (2019)
4. Holzinger, A., Carrington, A., Müller, H.: Measuring the quality of explanations: the system causability scale (SCS). KI-Künstliche Intelligenz **34**, 1–6 (2020)
5. Keane, M.T., Kenny, E.M.: How case-based reasoning explains neural networks: a theoretical analysis of XAI using *post-hoc* explanation-by-example from a survey of ANN-CBR twin-systems. In: Bach, K., Marling, C. (eds.) ICCBR 2019. LNCS (LNAI), vol. 11680, pp. 155–171. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-29249-2_11
6. Lamy, J.-B., Berthelot, H., Capron, C., Favre, M.: Rainbow boxes: a new technique for overlapping set visualization and two applications in the biomedical domain. J. Vis. Lang. Comput. **43**, 71–82 (2017)

7. Lamy, J.-B., Sekar, B., Guezennec, G., Bouaud, J., Séroussi, B.: Explainable artificial intelligence for breast cancer: a visual case-based reasoning approach. Artif. Intell. Med. **94**, 42–53 (2019)
8. Lipton, P.: Inference to the Best Explanation. Taylor & Francis, New York (2004)
9. Lipton, Z.C.: The mythos of model interpretability. Queue **16**(3), 31–57 (2018)
10. Massie, S., Craw, S., Wiratunga, N.: Visualisation of case-base reasoning for explanation. In: Proceedings of the ECCBR, pp. 135–144 (2004)
11. Molnar, C.: Interpretable Machine Learning. Lulu.com (2020)
12. Nugent, C., Cunningham, P.: A case-based explanation system for black-box systems. Artif. Intell. Rev. **24**(2), 163–178 (2005)
13. Ribeiro, M.T., Singh, S., Guestrin, C.: "Why should i trust you?" explaining the predictions of any classifier. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1135–1144 (2016)
14. Shapley, L.S.: A value for n-person games. Contrib. Theory Games **2**(28), 307–317 (1953)