GBEx – towards Graph-Based Explanations

Paweł Mróz*†, Alexandre Quemy*‡, Mateusz Ślażyński†, Krzysztof Kluza†, Paweł Jemioło†
* IBM Krakow Software Lab, Cracow, Poland
pawel.j.mroz@ibm.com, aquemy@pl.ibm.com
‡ Faculty of Computing, Poznań University of Technology, Poznań, Poland
† AGH University of Science and Technology al. Mickiewicza 30, 30-059 Krakow, Poland
{mslaz,kluza,pawljmlo}@agh.edu.pl

***Abstract*—This paper proposes Graph-Based Explanations (GBEx), a approach to explain machine learning models. It presents explanations in the form of a graph, where nodes represent arguments, and edges represent connections. The value of a graph node accounts for the influence of a given argument while the value of a graph edge accounts for the influence of a given connection. Contrarily to LIME, GBEx does not provide local explanations but a global explanations. And contrarily to SHAP, it can automatically explain interactions between variables. We provide an illustration on how GBEx can provide both local and global explanation.**

***Index Terms*—Explainable Artificial Intelligence, XAI, Graphs, Features**

I. Introduction

Nowadays, the most advanced machine learning algorithms are, for most part, black-box models. Consequently, there is no straightforward way to answer questions such as:

- Why, in this case, an algorithm made such a decision?
- What is the most important feature in the dataset?
- Does the model take irrelevant data into consideration?

To answer the above-mentioned questions, several tools have been developed the past years to make Artificial Intelligence (AI) more reliable [1], [2], [3]. With more complex and efficient models, usual performances metrics are not enough to trust a model. In addition, there is an increasing need for explainable models due to the impact of AI, e.g. in medicine [4] or the judicial system [5].

In this paper, we propose yet another method enabling interpretability models. The main contribution of the paper is a new approach relying on knowledge graphs for presenting explanations. We compare our method to the selected state of the art methods and provide preliminary validation on a real-life example.

II. State of the art

Interpretability is a measure of the degree to which persons can understand an output, and transparency, in turn, refers to the inherent internal features of the specific models, i.e., possibility to smoothly follow the process leading to generate the output or the results' presentation ease [1].

Black-box models create a gap between model explainability and performance [6]. Post-hoc explainability is regularly applied in the case of more sophisticated machine learning models. Thus, in such cases, additional techniques are added to the model for making the decision not only understandable but also justified. We distinguish two types of adapted strategies: model-agnostic (used to any machine learning model type), and model-specific (applied in correspondence with a selected learning tool) [1].

Local Interpretable Model-Agnostic Explanation (LIME) [7] is one of the most widely-used explainable techniques. Focusing on building linear models, it approximates and simplifies the mostly unintelligible outputs of the solutions.

This approach can explain black-box models with a variety of interpretable models and often with reasonable accuracy. As LIME works locally, one drawback is that there is a trust issue at the scale of the whole dataset. In particular, sensitive tasks require a more global explanation.

SHapley Additive exPlanations (SHAP) [8], based on game theory, measure the certainty calculated for each prediction based on the relevance of features in terms of the task. Unlike LIME, SHAP is fairly distributed among the whole dataset, so it is not only a local method.

Although SHAP has many advantages, it is computationally expensive and cannot automatically explain nonlinear factors or interactions between features. Moreover, in high dimensional space, the explanation can be confusing, and there is no way to reduce them, cnontrarily to LIME.

III. GBEx – Graph-Based Explanations

The main objective of Graph-Based Explanations (GBEx) is to represent explanations as graph, such that the value of a node in a graph represents the influence of a given argument and the value of an edge in a graph represents the influence of a given connection. Therefore, it forms an interpretable surrogate model that approximates the output from a black-box AI algorithm.

GBEx is made of two levels of abstraction. Most of the explanation models are focused on giving a certain value of importance to a specific feature or variable. In our approach, we add another layer of abstraction, which is to give a value of importance to a connection between specific arguments, therefore, enable automated explanation of two features or variable interactions.

GBEx models an explanation as follows:

$$\hat{y} = W^1\mu^1 + W^2\mu^2 + \beta \tag{1}$$

where:

- $\hat{y}$ – the vector to approximate or explain.
- W^1 – the matrix of inputs. Every case is represented by one row. 0 states the absence of an argument and presence is valued 1 divided by the number of present arguments in a specific case.
- μ^1 – the vector of nodes importance. Each value marks the influence of the corresponding argument to the output.
- W^2 – the matrix of connections. Similarly to inputs, each row represents some case. 0 means absence of connection and 1 divided by the number of present connections means that both arguments are occurring in a given case.
- μ^2 – the vector of edge importance, also similarly as for nodes this variable holds information about the influence of given connection to the output.
- β – the base value. When no arguments given, this is the predicted value.

In comparison, usual linear explanation methods rely on the following approximation model:

$$\hat{y} = W^1\mu^1 + \beta \tag{2}$$

There were quite a few challenges that needs to be addressed in order to obtain the different elements of Equation (1) that we present in the rest of this Section.

A. Binarization and clustering

The input matrix W^1 contains the information about the presence or absence of an argument in the dataset. However, most datasets contain non-binary features. In this paper, we distinguish between two types of data: categorical and real.

Categorical elements are simply one-hot encoded [9].

To handle real valued variables, we perform clustering using k-means [10]. According to the similarity metrics [11], [12], the clustering results are satisfying.

B. Solving equation

One approach consists in separating the task into two simple ones. First, we solve the following linear system, equivalent to a linear approximation as found in most other approaches:

$$\hat{y} = W^1\mu^1 + \beta^1 \tag{3}$$

Then the error that is left $e = \hat{y} - W^1\mu^1 - \beta^1$ is approximated in the same way by the second part:

$$e = W^2\mu^2 + \beta^2 \tag{4}$$

Another approach that would solve the equation at once is converting this problem into one larger linear system that can be solved simultaneously. That could be done by creating a matrix W^0 and vector μ^0 in the following way:

$$W^0 = [W^1 \;\; W^2] \tag{5}$$

$$\mu^0 = \begin{bmatrix}\mu^1\\ \mu^2\end{bmatrix} \tag{6}$$

The Equation 1 could be transformed to the following form:

$$\hat{y} = W^0\mu^0 + \beta \tag{7}$$

For most real problems, a direct matrix inversion is too computationally expensive. Because the matrix W^2 grows exponentially with the number of variables in the dataset, an iterative approach is preferred.

C. Presenting results

Presenting results is a crucial part of the explanations. The whole point of creating interpretable algorithms is to be able to show results in a human-friendly way. One of the primary goals of creating GBEx is to be able to present results in a graph form. One advantage of graphs over linear models for explainability is that it allows to naturally represent the interactions between variables while remaining easy to interpret.

As suggested by Equation (1), variables or features are represented by the nodes and relationships or interactions as edges. Additionally:

- Size – applied to node or edge. Represents the importance of an item.
- Color – applied to node or edge. Represents the class a feature or interaction supports. The difference in intensity of color points out to strength of the support.

The size and color intensity translates the same information: the strength of a variable or interaction in the decision toward a certain decision. This choice is motivated by the fact that it is easy to compare the importance of two opposing nodes by looking at size, rather than how intense is one color compared to another.

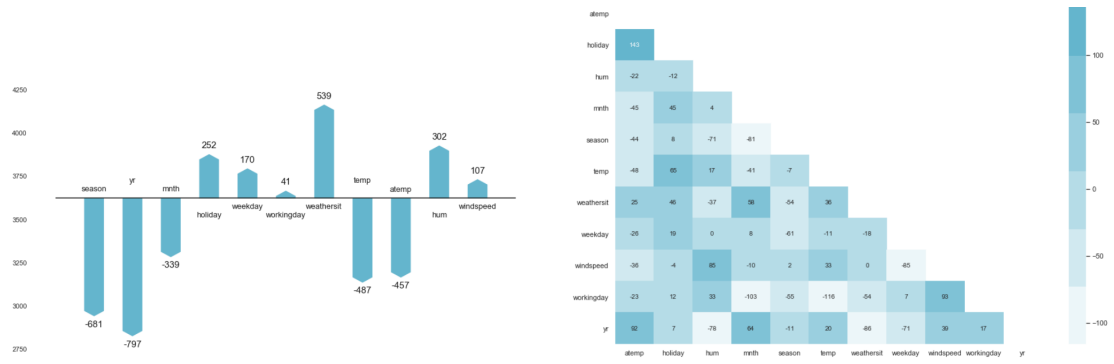
Depending on the number of clusters, the graph might become larger than one can comprehend. Therefore, we thus propose two ways of presenting explanations.

The first one is **general**. The explanation is focused on the whole dataset. To comprehend the amount of data presented, data from the same node are merged. Thus, if one feature was clustered into five groups, then the average of the importance of these fractions is taken. In that case, the knowledge about support for the given output is not really meaningful, and it is omitted. The second type of the explanation is **local** and focused on one specific case. The explanation is restricted to the features and interactions that appear a given case. The importance of features and interactions is re-normalized.

A graph seems the best way to present the importance of both features and binary relations between them in one picture. However, there are also other methods that could better display the parameters of the model separately.

Significance of an argument can also be presented at a simple bar chart. Such a structure shows clearly which feature is most important. It could easily plot the direction to which this argument is directing and how strongly. To plot only absolute importance, a pie chart could be used as well.

The same methods could be used to present explanations on a connection level. A clear and easy method to plot the dependency of connections is a heatmap. This structure was



(a) Feature. (b) Connection.

Figure 1. Influence in the final prediction for a given case.

Table I
DATASETS USED FOR VALIDATION AND ILLUSTRATION.

Dataset	Task	Cases	Features	Types
Bike rental	reg.	731	13	cat., bool., real
Skin	class.	245057	3	real
Heart diseases	class.	303	13	cat., bool., real

chosen because of its scalability. Even big connection matrices could be visualized by a heatmap in a very straightforward form.

IV. Validation and Illustration

To illustrate GBEx, we use two datasets from the UCI [13], one for regression and one for classification. Table I presents a description about each dataset.

A. One case explanations

We illustrate how GBEx mirror the output of model and provide parameters needed for the interpretation. To obtain the model, we used Multi-Layer Perceptron (MLP) Regressor.

The first approach is using a specific case and checking what contributed to the prediction in terms of features and interactions that exist between them.

We randomly took one vector to estimate the number of rentals. The prediction are as follows:

- Ground truth = 1011.
- MLP Regressor = 2026.
- Base value = 3624.
- GBEx = 1939.
- GBEx (without interactions) = 2274.

First, we observe that the prediction from the MLP is far from the ground truth. Nevertheless, the purpose of GBEx is not make predictions, but to explain a model and a decision, regardless if the model is good or not.

Another remark concerns the value obtained with and without the interactions, which is the main criterion differentiating GBEx from other method. We observe that without the interactions, distance from the MLP Regression prediction is larger than if we take them into account. This comfort us in the fact that graphs provides a richer structures to explain the model.

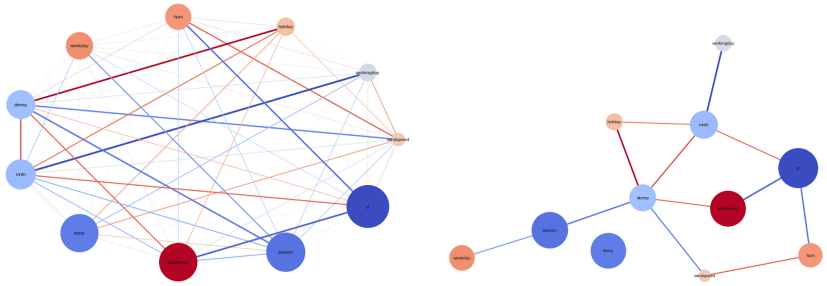
The influence of variables in the final prediction is shown in Figure 1(a). Note that the illustration does not take into account the interactions. The starting point of this plot is a base value, which is set to be 3624 bikes rented. According to the plot, the most influential features are 'Season' and 'Year'. Understandably, the earlier year is, the less active users there are. Moreover, winter is arguably the least friendly season for bike riding. On the other hand, the most persuasive opposing feature was the 'weather situation'. Other important factors are 'temperature' and 'feeling temperature'. As would be expected in December, they also contribute to the lowering of the final prediction. An unusual thing could be noticed when checking the 'holiday' parameter. In this case, the day was marked as non-holiday. Understandably, it was supporting to increase the score, but the day was 24.12.2011, which is Christmas Eve. That might be, at least partially, explaining not so accurate prediction made by MLP.

Te give yet another representation of the interactions strength, we display the heatmap in Figure 1(b). It shows how a given coalition is contributing to the final prediction.

The picture shows that one of the most important interactions was between the fact that it was a non-holiday day, and the temperature was low. Other significant interaction are those between 'working day' and 'month' or 'temperature'. It might be understood as lowering predictions for a working day when it is December, which make sense since, December is cold and during working days, people might prefer commuting using a warmer way of transportation.

So far, we represented the features influence and interactions influence separately. However, it is not simple to grasp the whole explanation from these two separate analysis. That is why we display the explanation graph in Figure 2(a). The similar case is used as in the previous example.

As there are an exponential amount of possible interactions in function of the number of nodes, it can be difficult to read the fully connected graph, even with few nodes. To simplify the graph, we cut off the least important edges and present only those whose influence is most significant. We used a predefined threshold to delete less relevant connections and nodes. The simplified graph is presented in Figure 2(b). It still



(a) Full. (b) Simplified.

Figure 2. Explanation graph, showing the general importance of variables and interactions in the model.

contains most of the useful information and is much clearer and more readable.

B. General explanation

One advantage of GBEx is that explanations are made not only at a specific case level: it also contains useful information about the whole model. As mentioned earlier, to keep clarity, there is a need to merge the arguments coming from the same feature. Merging is done by taking the average absolute value of the influence of arguments.

Figure 2(b) shows the importance of variables in the decisions over the whole dataset $bike$. We can observe that the most influential feature was 'year' as it was responsible, on average, for 21% of the whole explanation coming from nodes. It is followed by 'weather situation' 'temperature' and 'season' which seem to be reasonable factors to explain the variation in a the number of bikes rented. Finally, the arguments that contributed the least were 'working day', 'weekday', and 'holiday'. The disadvantage of this approach is the inability to show the direction in which arguments are supporting.

On top of the individual contribution of each variable, we displayed the interactions contributions in the heatmap in in Figure 3(b). According to this heatmap, the most influential connection was the one between 'working day' and 'month'. Other important edges were those connecting 'year' with 'humidity' and 'weather situation'.

As would be expected weather arguments have a lot of influence. Given that the popularity of the bike-sharing business is growing with time, it is also understandable that year was the most important feature. Least important are columns related to the day of week or holiday, which is reasonable given that those are not key factors when deciding whether to rent a bike, compared to weather conditions.

Again, these results seem to reasonably explain how the predictions are made on such naive datasets which comfort us in the capability of GBEx to handle more complex situations.

General explanations also provide a possibility to present data in a graph form. But given a lack of ability to provide discriminatory support, the color of nodes does not hold any information. The strength of influence is represented by the size of nodes as well as edges.

C. Feature analysis

As for real values a clustering algorithm have been applied, in this section, we analyze how the explanation changes depending on the number of clusters. The question we try to answer is: how the number of cluster influence interpretability?

For this test, we used $skin$ dataset.

Figure 4(a) presents the influence of each of five clusters made from the third feature. The group with the higher number represents the largest values. The red dashed line marks the neutral point, i.e., the feature supports neither positive nor negative outcome. As this is a classification task, the odds are presented on a logarithmic scale. The blue line points out the base value generated by the logistic regression. The dependency seems to be clear: the larger the variable, the more support toward the positive outcome. On the other hand, Figure 4(b), show slightly different results. The middle picture, with 10 clusters, presents similar behavior as for a smaller number of clusters, but on the sides, we notice that the behavior is reversed. We explain this phenomena by the fact that with too few clusters, the average support within a cluster hides important information.

Figure 4(c) shows the support of the same feature divided into 15 clusters. The shape created by these values is similar to the previous ones, with the same effect on the sides.

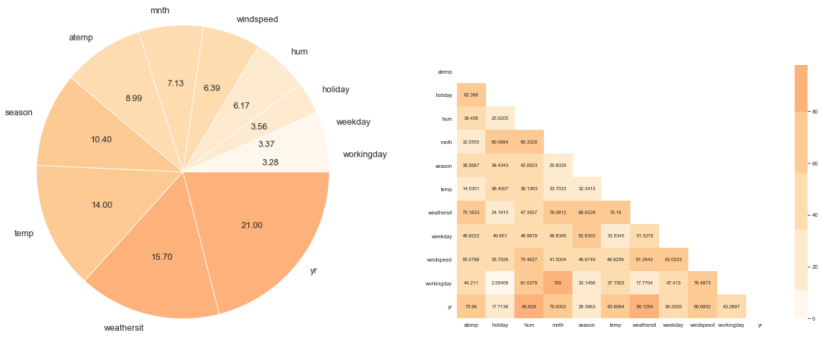
D. Performance

To be adopted in practice, an algorithm does not only need to perform well, but also to be fast enough to tackle the problems. As mentioned before, the complexity of GBEx is exponential in the number of dimensions of the input vector space. Fortunately, fast iterative algorithms are available and can provide an approximate solution in a reasonable time even for large instances.

As an illustration, we checked time needed to run the algorithm on $skin$ dataset. As expected, the time is growing exponentially according to the amount of data (Figure 5(a)) taken into the computation and nearly exponentially with the number of clusters (Figure 5(b)).

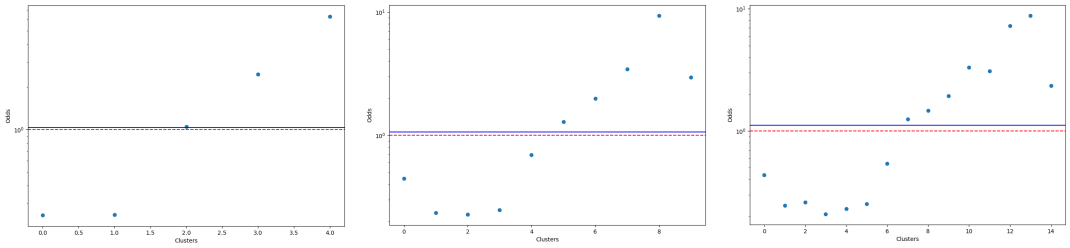
V. Conclusions and Future Work

In this paper, we proposed Graph-Based Explanations, a new method for creating a surrogate model to interpret decisions made by any model. The explanations are presented



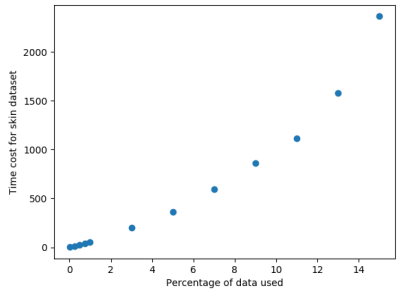
(a) Argument. (b) Edges.

Figure 3. Importance in general model.

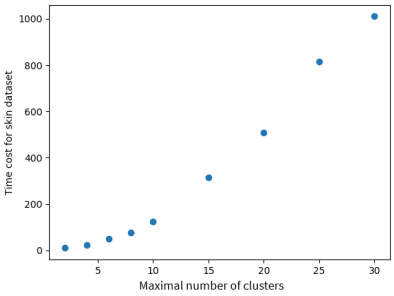


(a) 5 clusters. (b) 10 clusters. (c) 15 clusters.

Figure 4. Support of third feature from $skin$ dataset.



(a) Amount of data as variable.



(b) Number of clusters as variable.

Figure 5. Time needed to produce the explanations [seconds].

as a graph, where arguments are represented as nodes and connections by edges.

Contrarily to previous methods such as LIME or SHAP, the output of the model to explain is not modeled by a simple hyperplan but by a graph. This structure allows us to account for any binary interactions, thus, improving the explainability power while remaining simple enough for visualisation and interpretation.

We demonstrate how to obtain an explanation, both for a specific case and for the whole model. We showed on simple real life examples that GBEx provides a satisfying explanations for the most influential variables and interactions. Although it is possible to make the visualization in the form of a graph, arguments and interactions can be analyzed separately.

The downside of a richer structure of explanation like a graph is that the algorithm does not scale to extremely large datasets due to the size of the linear system to solve that grows exponentially. Therefore, it is crucial to increase its efficiency, for instance by pruning less relevant cases or features during the algorithm run. It would be also beneficial to develop and include methods for deciding the optimal number of clusters. For now, the choice is left to the user and might require a bit of feature analysis. Despite the fact that it is probably not possible to find a universal solution to all scenarios, some compromises can probably be done automatically.

Last but not least, a more systematic analysis of GBEx explanation is needed, as well as a proper comparison with LIME and SHAP.

References

[1] A. B. Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. García, S. Gil-López, and D. Molina, "Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI," *Information Fusion*, 2019.

[2] A. Adadi and M. Berrada, "Peeking inside the black-box: A survey on explainable artificial intelligence (XAI)," *IEEE Access*, vol. 6, pp. 52 138–52 160, 2018.
[3] C. Molnar, "Interpretable machine learning: A guide for making black box models explainable," 2018.
[4] L. Wang and A. Wong, "COVID-Net: A tailored deep convolutional neural network design for detection of covid-19 cases from chest radiography images," *arXiv preprint arXiv:2003.09871*, 2020.
[5] A. Deeks, "The judicial demand for explainable artificial intelligence," *Columbia Law Review*, vol. 119, no. 7, pp. 1829–1850, 2019.
[6] F. K. Došilović, M. Brčić, and N. Hlupić, "Explainable artificial intelligence: A survey," in *2018 41st International convention on information and communication technology, electronics and microelectronics (MIPRO)*. IEEE, 2018, pp. 0210–0215.
[7] M. T. Ribeiro, S. Singh, and C. Guestrin, "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*. ACM, 2016, pp. 1135–1144.
[8] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Advances in Neural Information Processing Systems*, 2017, pp. 4765–4774.
[9] R. Vasudev, "What is one hot encoding? Why and when do you have to use it?" 2017, https://medium.com/hackernoon/what-is-one-hot-encoding-why-and-when-do-you-have-to-use-it?
[10] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine learning in python," *the Journal of machine Learning research*, vol. 12, pp. 2825–2830, 2011.
[11] D. L. Davies and D. W. Bouldin, "A cluster separation measure," *IEEE transactions on pattern analysis and machine intelligence*, no. 2, pp. 224–227, 1979.
[12] P. J. Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis," *Journal of computational and applied mathematics*, vol. 20, pp. 53–65, 1987.
[13] D. Dua and C. Graff, "UCI machine learning repository," 2017, http://archive.ics.uci.edu/ml.