

Efficient and simple prediction explanations with groupShapley: A practical perspective

Martin Jullum¹, Annabelle Redelmeier¹ and Kjersti Aas¹

¹Norwegian Computing Center, P.O. Box 114, Blindern, N-0314 Oslo, Norway

Abstract

Shapley values has established itself as one of the most appropriate and theoretically sound frameworks for explaining predictions from complex machine learning models. The popularity of Shapley values in the explanation setting is probably due to Shapley values' unique theoretical properties. The main drawback with Shapley values, however, is that the computational complexity grows exponentially in the number of input features, making it unfeasible in many real world situations where there could be hundreds or thousands of features. Furthermore, with many (dependent) features, presenting/visualizing and interpreting the computed Shapley values also become challenging. The present paper introduces and showcases a method that we call *groupShapley*. The idea of the method is to group features and then compute and present Shapley values for these groups instead of for all individual features. Reducing hundreds or thousands of features to half a dozen or so feature groups makes precise computations practically feasible, and the presentation and knowledge extraction greatly simplified. We give practical advice for using the approach and illustrate its usability in three different real world examples. The examples vary in both data type (regular tabular data and time series), feature dimension (medium to high), and application (insurance, genetics, and banking).

Keywords

Shapley values, prediction explanation, feature grouping, computational feasibility

1. Introduction

Consider a predictive modelling/machine learning setting with a model $f(\cdot)$ that takes an M dimensional feature vector $\mathbf{x} = \{x_1, \dots, x_M\}$ as input and provides a prediction $f(\mathbf{x})$ of an unknown response y . Suppose that for a specific feature vector, $\mathbf{x} = \mathbf{x}^*$, we want to understand how the different features (or types of features) contribute to the specific prediction outcome $f(\mathbf{x}^*)$. This task is called prediction explanation and is a type of *local* model explanation, as opposed to a *global* model explanation, which attempts to explain the full model at once, through concepts such as global feature importance [1, Ch. 2].

Shapley values [2] is a leading framework for prediction explanation. The methodology has, in particular, received increased interest following the seminal XAI paper of Lundberg and Lee [3]. The feature-wise Shapley values ϕ_1, \dots, ϕ_M for a predictive model $f(\mathbf{x})$ at $\mathbf{x} = \mathbf{x}^*$ are

XAI.it 2021 - Italian Workshop on Explainable Artificial Intelligence

✉ jullum@nr.no (M. Jullum); anr@nr.no (A. Redelmeier); kjersti@nr.no (K. Aas)

🌐 martinjullum.netlify.app (M. Jullum)

🆔 0000-0003-3908-515 (M. Jullum)



© 2021 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings (CEUR-WS.org)

given by

$$\phi_j = \sum_{\mathcal{S} \subseteq \mathcal{M} \setminus \{j\}} \frac{|\mathcal{S}|!(M - |\mathcal{S}| - 1)!}{M!} (v(\mathcal{S} \cup \{j\}) - v(\mathcal{S})), \quad (1)$$

for $j = 1, \dots, M$, where \mathcal{M} is the set of all features, and $v(\cdot)$ is a characteristic/contribution function. $v(\cdot)$ represents an expected prediction knowing only feature values in \mathcal{S} , for which there exists different definitions and estimation procedures that are appropriate in different situations. The motivation and methodology of the present paper is, however, independent of the choice of $v(\cdot)$. We therefore leave it unspecified for now. See e.g. [4, 5, 6] for details and discussions of this topic, and Section 4 for the definitions and methods used in the real data examples of the present paper. On the interpretability side, the Shapley value ϕ_j corresponds to the change in the prediction caused by observing feature j – averaged over knowledge of the other features.

By inspecting (1), it is clear that the sum in the Shapley value formula grows exponentially in the number of features, M , and thus becomes computationally intractable when there are large number of features. This problem has received a lot of attention both from the general game theoretic side [7, 8, 9, 10], and the specific machine learning/XAI setting we are considering [3, 11, 12, 13, 14, 15]. However, these methods are either restricted to certain models, require strong assumptions, or trade speed for approximation accuracy and thus eventually become too imprecise as the number of features increases. In particular, both the TreeSHAP and KernelSHAP method of [3, 15], which are heavily used within XAI, belong to the third category [16]. In any case, with hundreds or thousands of computed Shapley values, the presentation/visualization, interpretation, and knowledge extraction become extremely difficult. This is especially the case when some of the features are (highly) dependent, as their joint contribution to the prediction function is (rightly so) spread out on the different features. This results in many small Shapley values, see e.g., [4]. In most applied fields, feature dependence is the rule rather than the exception. Thus, feature-wise prediction explanation using Shapley values remains notoriously difficult. The question is then whether there is a way to obtain simple and intuitive explanations of predictions from models with large number of features through the well-founded framework of Shapley values.

1.1. The present approach

The present paper introduces and showcases *groupShapley*, a conceptually simple method that explains predictions using the Shapley value framework for groups of features rather than individual features. The paper is based on a more comprehensive and technical unpublished paper [17], which, to the best of our knowledge, was the first to propose this approach for prediction explanation. *groupShapley* simply replaces the individual features in (1) by feature groups, creating perfectly well-defined Shapley values with all of the usual Shapley value properties. Since the summation is over group sets rather than feature sets, the computational complexity is kept small and the presentation/visualization of the Shapley values is no longer an issue, as long as the number of groups is small. The way the features are grouped has important implications for the interpretability of the computed Shapley values and the knowledge extraction thereof.

We advocate to group the features in a practically meaningful way, where similar feature types are grouped together. This simplifies knowledge extraction and gives Shapley values with a practical interpretation. For instance, in the case where a few base features/data sources are used to construct a myriad of engineered features used by the model, *groupShapley* can provide simple interpretations of predictions from this model by only presenting Shapley values for the original base features or data sources.

Finally, *groupShapley* is implemented in the `shapr` R-package [18]¹, allowing for simple computation and visualization of explanations based on any application and predictive model.

The rest of the paper is organized as follows: Section 2 provides a brief overview of other work on Shapley values for groups/coalitions and related approaches. Section 3 provides the mathematical definition of *groupShapley* and discusses how groups can be constructed in different practical settings. Then Section 4 gives three real world examples using car insurance, genetics, and time series data sets. Finally, Section 5 provides a brief summary and some concluding remarks.

2. Related work

In the general game theoretic literature, various notions of Shapley values with a grouping component have been proposed and studied. Papers such as [19, 20, 21, 22] study various variants of so-called quotient games and coalition games. However, such concepts do not reformulate the game of interest like we do with *groupShapley*, but rather bring in the coalitional structure as an additional component to the original game. Also, most of these formulations give values to the original players and not the groups themselves. [23, 24] study and quantify the cooperation strength between two or more players by defining a Shapley interaction index. [25, 26, 27] study the profitability of forming coalitions/groups for individual players in a general game theoretic setting. Although the computed value is termed the ‘Shapley group value’ in [25], it has no direct connection to *groupShapley* in the present paper.

As mentioned in Section 1, there are also several papers that aim at reducing the computational burden of calculating Shapley values for the original players. Some of these utilize grouping structures to achieve this, see e.g., [7, 8, 28].

In addition to the previously mentioned paper [17], the formula used by *groupShapley* has independently and very recently appeared in preprints [29, 30, 31, 32]. Although all of these papers treat the *groupShapley* formula to some degree, they have different motives and are rather mathematical heavy, making them inaccessible for many practitioners. Instead, this paper takes a practical application oriented perspective. We minimize the mathematical details, gear the group construction towards explanations that are useful for the specific application, and showcase this on several relevant use cases.

Finally, note that [4, Sec 5.] suggests to use sums of feature-wise Shapley values as grouping scores. However, these scores are not, in general, Shapley values, and do not bypass the computational issue, see also [17, 31] for comparisons.

¹The *groupShapley* methodology is (as of November 2021) only available in the development version of the package on GitHub: github.com/NorskRegnesentral/shapr.

3. groupShapley

Recall the predictive modelling/machine learning setting from Section 1 with the predictive model $f(\cdot)$ that produces predictions based on an M -dimensional feature vector \mathbf{x} that one wishes to explain. Let us now define a partition $\mathcal{G} = \{\mathcal{G}_1, \dots, \mathcal{G}_G\}$ of the feature set \mathcal{M} , consisting of G groups². Then the Shapley value for the i -th group (\mathcal{G}_i) explaining the prediction $f(\mathbf{x}^*)$ is given by

$$\phi_{\mathcal{G}_i} = \sum_{\mathcal{T} \subseteq \mathcal{G} \setminus \mathcal{G}_i} \frac{|\mathcal{T}|_g!(G - |\mathcal{T}|_g - 1)!}{G!} (v(\mathcal{T} \cup \mathcal{G}_i) - v(\mathcal{T})), \quad (2)$$

where the summation index \mathcal{T} runs over the groups (not the individual features) in the set of groups $\mathcal{G} \setminus \mathcal{G}_i$, and $|\mathcal{T}|_g$ refers to the number of groups (not the individual elements) in \mathcal{T} . The contribution function $v(\cdot)$ for groups of features can be directly extended from those for individual features.

As *groupShapley* is simply the game theoretic Shapley value framework applied to groups of features instead of individual features, the *groupShapley* values possess all the regular Shapley value properties. In particular, the *efficiency property* states that $\sum_{i=1}^G \phi_{\mathcal{G}_i} = f(\mathbf{x}^*) - v(\emptyset)$, the *symmetry property* roughly states that groups which contribute equally (regardless of the influence of other feature groups) have identical Shapley values, and the *null player property* states that feature groups with no contribution to the prediction (neither directly, nor through other features) have a Shapley value of zero.

By directly comparing the formula for the feature-wise Shapley values in (1) with the *groupShapley* formula in (2), we see that the computational complexity of the sum reduces from 2^{M-1} to 2^{G-1} , which gives a relative computational cost reduction of 2^{M-G} . With, for example, $M = 50$ features and $G = 5$ feature groups, the relative cost reduction is $2^{50-4} > 10^{13}$. That is, the computational cost reduction is huge when moving from feature-wise Shapley values to *groupShapley*.

3.1. Defining feature groups

With the general *groupShapley* procedure established in (2), the next task is defining feature groups. We outline some suggestions below.

Feature knowledge: Grouping of features based on underlying knowledge depends on the problem at hand. With typical, medium sized tabular data, one can create groups intuitively. When predicting housing prices, for example, groups can be *house attributes* (square footage, building materials), *luxury amenities* (presence of hot tub, pool, home theatre), *locality* (distance to nearest grocery store/bus stop/gym), *environmental factors* (crime rate, air pollution, traffic, sun exposure), and *historical turnover*. See also the car insurance example in Section 4.1. If less is known about the features or the feature dimension is too large to manually create groups, predefined groups based on domain knowledge can be used. See our gene example in Section 4.2. Time series data can be grouped according to time – for example all values belonging to specific

²I.e. a grouping of features where each feature is represented in exactly one group and no group is empty.

years, months or weeks depending on the time scale and required explanation resolution. See also the example in Section 4.3.

Feature dependence: We can also group features based on dependence. These groups can be found using a clustering method that uses a dependence based similarity measure, see e.g., [33] and [4, Sec. 5]. A benefit of this type of grouping is that it is not connected to the specific application at hand, making it is easier to study theoretically. See [17] and [31] for work in that direction.

The advantage of grouping based on feature knowledge and not feature dependence is that the resulting Shapley values provide a directly meaningful interpretation for the specific model prediction. Such an approach also allows groups to be formed based on the application and problem at hand. On the other hand, dependence based grouping provides groupings without practical meaning that are difficult to learn from. In the following section we focus only on groups based on feature knowledge. This is our preferred grouping strategy, and allows us to compare the *groupShapley* values directly with insights from the data sets.

3.2. Remarks

An obvious consequence of using *groupShapley* rather than computing feature-wise Shapley values, is that one gets a less detailed explanation and cannot generally draw conclusions on the contribution of individual features. While this might be a significant drawback in some settings, we believe that it is a strength in most situations, as it keeps the explanations simple and forces the user to focus on the bigger picture. As mentioned in 1, extracting knowledge from hundreds or thousands of feature-wise Shapley values is inherently difficult, but is much easier based on only a few *groupShapley* values. Furthermore, if one is particularly interested in the contribution from a few specific features, these could be kept as is, while the remaining features are grouped.

It is natural to ask whether there is a link between feature-wise Shapley values and *groupShapley* values. [17, 31] studies various aspects of this when the grouping is based on feature dependence. However, to the best of our knowledge, there exists no general results that allows computing feature-wise Shapley values from *groupShapley* values or vice versa. It may be worth noting that by re-applying the Shapley framework to decompose a *groupShapley* value into the features of the group, one obtains so-called Owen values for the features, see [21] for further details.

A clear requirement for using *groupShapley* is that one is able to divide the features into groups that makes sense. If the user has no knowledge about the features at all, and the dependence structure of the features does not provide a meaningful way of clustering them, then *groupShapley* is not a viable direction. However, not only are such scenarios rare, we also believe that it is not very useful to explain predictions in such scenarios, since the complete lack of information makes it impossible to transform any explanations to useful knowledge.

4. Real data examples

In this section, we demonstrate how *groupShapley* can be applied to three different real data settings. In all of the examples, we use the conditional expectation as contribution function, i.e.,

$$v(\mathcal{T}) = \mathbb{E}[f(\mathbf{x}) | \mathbf{x}_{\mathcal{T}} = \mathbf{x}_{\mathcal{T}}^*], \quad (3)$$

where $\mathbf{x}_{\mathcal{T}}$ denotes the subvector of \mathbf{x} corresponding to feature subset \mathcal{T} . This contribution function accounts properly for the dependence between the features [5], both within and between the groups. For feature-wise Shapley values, (3) was introduced by [3], and has since been used by several others [4, 6, 16, 34]. As mentioned in Section 1, the contribution function needs to be estimated. In this paper we use Monte Carlo integration to approximate the conditional expectation, see [16] for details. In that approach one needs an estimate of the corresponding conditional distribution. Due to the different types of data in the three examples, we will use three different methods to estimate conditional distributions, see the respective subsections. All of the *groupShapley* computations were produced by our R-package *shapr*.

Note that these examples are for illustrational purposes only. A thorough explanation of the predictive models, the mechanisms between the features, and the response in each application require both a thorough analysis of the appropriateness of the fitted models and careful selection of methods for modelling the feature distributions. This is beyond the scope of this paper. We keep the explanations of the fitted models simple and report performance through area under the receiver operator curve (AUC) [35] and Brier score [36].

4.1. Predicting car insurance risk with customer demographics

In this first example, we use a car insurance data set found on the Kaggle website. The data contains two different response variables, 23 features, and 10,302 observations. The response we use is the binary variable *customer had a claim*. The features can be naturally partitioned into the following groups:

- *Personal Information*: age of driver, highest education level, number of children living at home, value of home, income, job category, number of driving children, marital status, single parent, gender, distance to work, whether driver lives in an urban city, how many years driver has had job.
- *Track Record*: number of claims in the past five years, motor vehicle record points, licence revoked in past seven years, amount of time as customer.
- *Car Information*: value of car, age of car, type of car, whether car is red.

Five of the variables have missing data. We use predictive mean matching to impute these. To model the probability of a claim, we use 90% of the data to train a random forest model with 500 trees using the *ranger* R-package [37] on the binary response and all 23 features. On the remaining 10% of the data, we get an AUC score of 0.835, and Brier score of 0.148. The average predicted probability of a claim is 0.273. We then use *groupShapley* to calculate Shapley values for the *Personal Information*, *Car Information*, and *Track Record* groups for four different individuals. Since there is a mix of continuous, discrete, and categorical features, the conditional

inference tree approach of [6] is used to estimate the required conditional distributions used in the Monte Carlo integration to estimate (3). We plot the three grouped Shapley values for four different individuals in Figure 1.

The first individual is a single mother of four (where two children drive). She drives an SUV and drives 27 miles to work. She has had one claim in the last five years and has three motor vehicle record points. *Personal Information* gives the largest increase in the predicted probability, which is not surprising given her travel distance and two young drivers. The second individual is a 37-year-old father of two (where one child drives). He has had one claim in the last five years, his licence revoked in the last seven years, and ten motor vehicle points. His *Track Record* significantly increases his predicted probability, which is natural given his misdemeanors.

The third individual is a 60-year-old married male with no kids at home. He drives a red sports car and has had three claims in the last five years. He has a PhD and currently works as a doctor. His *Personal Information* naturally reduces his predicted probability, while his poor *Track Record*, and to some extent his luxurious *Car Information* increases his predicted probability. The fourth individual is a 50-year-old female with no kids at home. She drives a minivan and has no previous claims or revoked licences. She also has a PhD and drives 42 miles to work. She appears to be on the safer side of things, which is reflected in all negative *groupShapley* values and a smaller predicted probability.

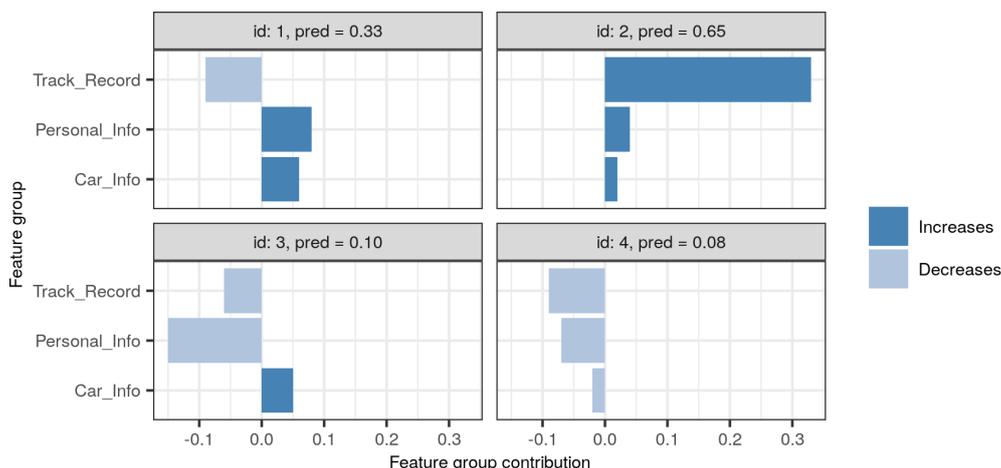


Figure 1: Estimated *groupShapley* values for four individuals in the car insurance example, with the groups defined in Section 4.

4.2. Predicting inflammatory bowel diseases with genes

In this section, we show how *groupShapley* can be used with high dimensional gene data when the goal is to decide how groups of genes influence the prediction for different types of patients. We use a data set from [38] with 127 patients: 85 with inflammatory bowel disease (IBD) where 59 have Crohn's disease (CD) and 26 have ulcerative colitis (UC), and 42 healthy controls. Each patient has data for 22,283 genes run on an affymetrix U133A microarray [38]. The example in this section is motivated by an application in [39] which uses this data set to motivate the use

of sparse-group lasso³.

We begin by modelling the probability of falling in either diagnosis class (IBD = CD or UD) versus the probability of being a control. To really challenge *groupShapley*, we here use the approach to see if we can identify differences in the explanations between the patients with CD versus the patients with UC, despite the model not knowing the difference.

Since calculating Shapley values on more than 22,000 genes is both infeasible and useless (most would probably be close to zero), we group the genes using the Hallmark gene sets [40]. This reduces the 22,283 genes to 50 gene sets. Analogously to [39], we remove the genes that are not part of any of the gene sets, leaving us with 6,965 genes. We then choose 100 patients at random to use as our training set and produce explanations of the predictions for the remaining 27 individuals.

We use the `glmnet` R-package to fit a penalized logistic regression model with L_1 regularization (Lasso) [41] to these data and use 10 fold cross-validation based on binomial deviance to select the shrinkage parameter. This leaves us with 45 non-zero regression coefficients belonging to 23 different gene sets. Due to the potential dependence between genes in the same gene set, we consider all genes (4834 in total) contained in these 23 gene sets.

Although computing *groupShapley* for 23 gene sets is several magnitudes simpler than computing Shapley values for thousands of individual genes, it is still computationally heavy. We therefore use the KernelSHAP approach of [3] to reduce the sum in (1) to 5000 group subsets, at the cost of some approximation error. Since the features (genes) in this example are all continuous, and their high dimension makes it difficult to evaluate specific distributional assumptions, we rely on the nonparametric *empirical* method of [16] to approximate the required conditional distributions used in the Monte Carlo integration to estimate (3).

In this simplified binary classification problem, we achieve perfect discrimination between diseased and control individuals in our test set with an AUC of 1. The Brier score is 0.004. Since for illustrational purposes we concentrate on how the explanations vary *between* different patient classes (healthy controls, CD, and UC), we show box plots of the estimated *groupShapley* values for each gene set and patient class, instead of the *groupShapley* values per individual. These box plots are given in Figure 2. We see that the explanations unsurprisingly vary significantly between the controls and diagnosed, but that there also are some differences between UC and CD.

Below we identify some studies that back up some of the differences in *groupShapley* values between controls, UC, and UD patients in Figure 2. Starting from the top of Figure 2, the clearest distinction between the *groupShapley* values for healthy controls and CD/UD (with a negative value for the former and a positive for the latter) is present for the ‘P53 Pathway’ gene set. The importance of the P53 gene set is also confirmed in the meta study [42] and references therein.

Second, the ‘TNFA signaling via NFKB’ gene set contains the ‘NFKB2’ and ‘COPEB (KLF6)’ genes which are highlighted in [38] as genes that can distinguish between UC and CD. This gene set is also where we see the largest difference in the *groupShapley* box plots between UC and CD. A pure CD versus control study [43] concludes that the ‘TNFA signaling via NFKB’ gene set has an important role for characterising CD. This can also be seen in Figure 2. Gene set 5 from

³Although we follow their data processing steps, we cannot compare *groupShapley* to sparse-group lasso as the latter is a predictive model, not an explanation method.

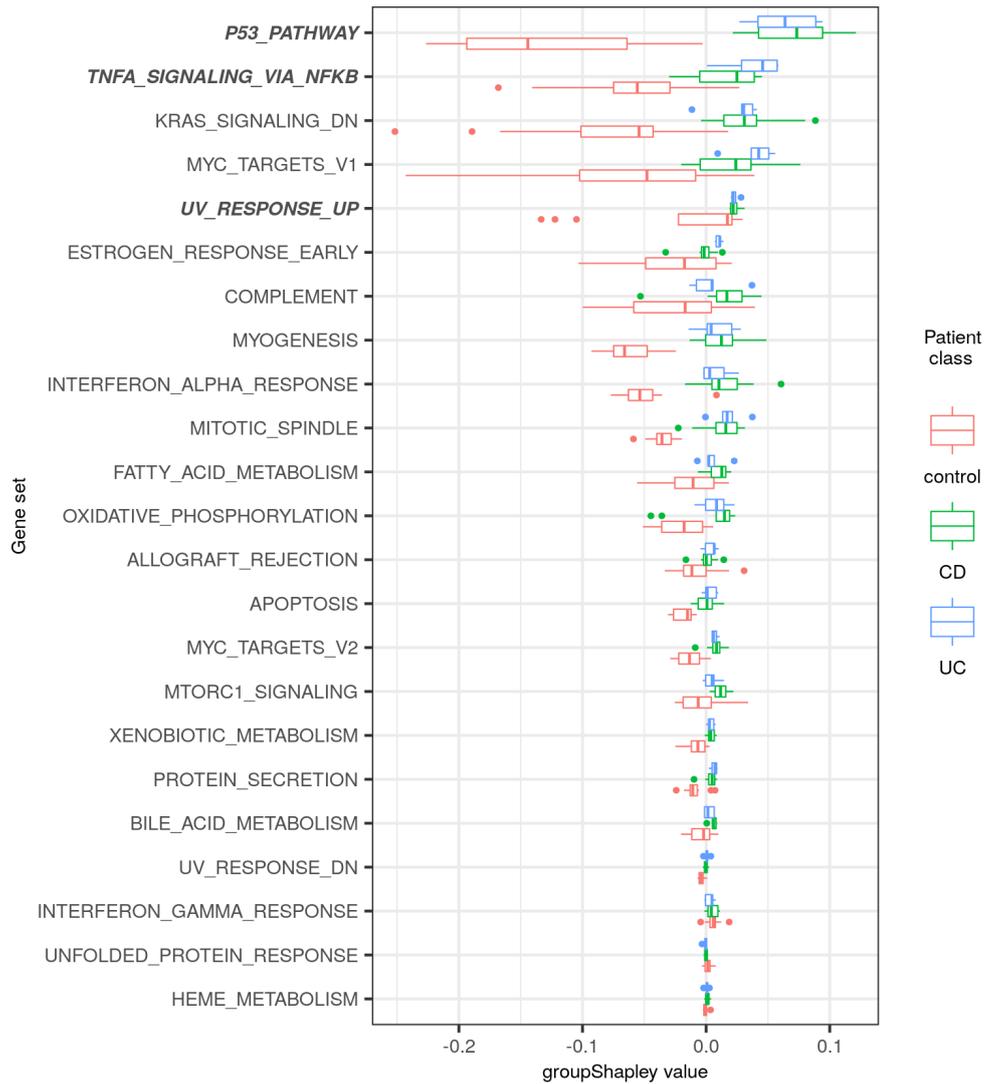


Figure 2: Box plots of estimated *groupShapley* values for 27 individuals, per patient class in the IBD gene example, using the Hallmark gene sets as groups. Gene sets discussed are given in bold italics.

the top, ‘UV response up’, is also mentioned by [43] as a potential identifier of CD compared to controls, which is also indicated by the big differences between CDs and some of the controls in Figure 2. While our box plots also indicate interesting differences for some of the other gene sets, we have not been able to find support for that in the biological/medical literature. However, we are not domain experts, and have not carried out an exhaustive literature review on CD/UD in relation to the Hallmark gene sets, so such support could also exist.

This example serves as an illustration of how *groupShapley* can be used to extract knowledge and explain models based on thousands of genes. While we used a general purpose gene set, knowledge extraction may increase if these groupings are specially prepared to the specific application by domain experts.

4.3. Predicting mortgage default with time series data

This example is based on the the work of [44] which predict future mortgage default based on customers' bank accounts. The data comes from Norway's largest bank (DNB) and includes 20,989 customers which previously had a mortgage between 2012 and 2016. In [44], separate convolutional neural networks (CNN) are fitted to six time series of daily records from the customers' bank accounts to predict the probability of default within a specified future time window. We refer to [44] for details about the model specification, data preparation, and modelling setup. To simplify the present example, we only consider one of these time series, consisting of the sum of the daily account balances of the checking account, saving account, and credit card account. Using only this single time series, their CNN model achieves an AUC of 0.867 and a Brier score of 0.044 on their out-of-sample and out-of-time test set of 1921 customers.

To exemplify how *groupShapley* can be used on time series data, we divide the input time series into four non-overlapping time periods: Q1-Q4. We then compute *groupShapley* values for each of these quarters to understand how each part of the time series contributes to the prediction. A figure showing the time series of four customers⁴ and their respective predictions are shown in Figure 3. The four quarters are shaded respectively.

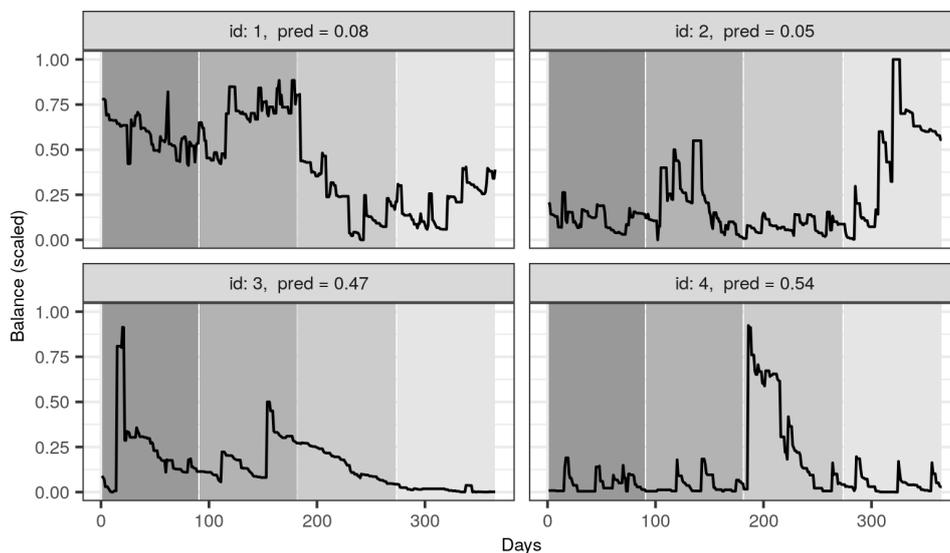


Figure 3: Time series plots of bank account balances of four fictive individuals in the mortgage default prediction example. The balances are scaled to the interval $[0,1]$, see [44] for details. The different greyscales indicate the different quarters of the year.

There have been surprisingly little focus on explaining predictive models based on time series data with Shapley values. To explain predictions from this model, we therefore need new methodology for estimating the conditional distributions required in the Monte Carlo integration to estimate (3) that appropriately accounts for the time/ordering component. Since this is a research topic on its own, we here settle for a simple approach which has conceptual

⁴Since these account balances contain sensitive information, this is simulated data based on the accounts of real customers.

similarities with the *empirical* method of [16]. The approach is briefly described below.

Since the subsets \mathcal{T} correspond to one or more of the quarters Q1-Q4, we need to estimate quantities like the expected prediction when all values except e.g., those in Q2 are known, i.e. $v(\{Q1, Q3, Q4\}) = \mathbb{E}[f(\mathbf{x})|\mathbf{x}_{Q1, Q3, Q4}]$. By once again relying on Monte Carlo integration, we need samples from (an approximation of) the conditional distribution $p(\mathbf{x}_{Q2}|\mathbf{x}_{Q1, Q3, Q4})$. We approximate this distribution by sampling from the training data and then adjusting these samples so that they align with the individual we want to explain. To explain this adjustment we refer to Figure 4: The endpoints of the sampled series (the green series) will not connect with the end points of our individual's series. To make this happen, we take the difference between the sampled series and the line connecting the end points (dashed green line) and add this difference to the line connecting the endpoints of our individual's series (dashed red line). This adjusted series is shown as the bold blue line. The entire bold curve is used when estimating $v(\mathcal{T})$.

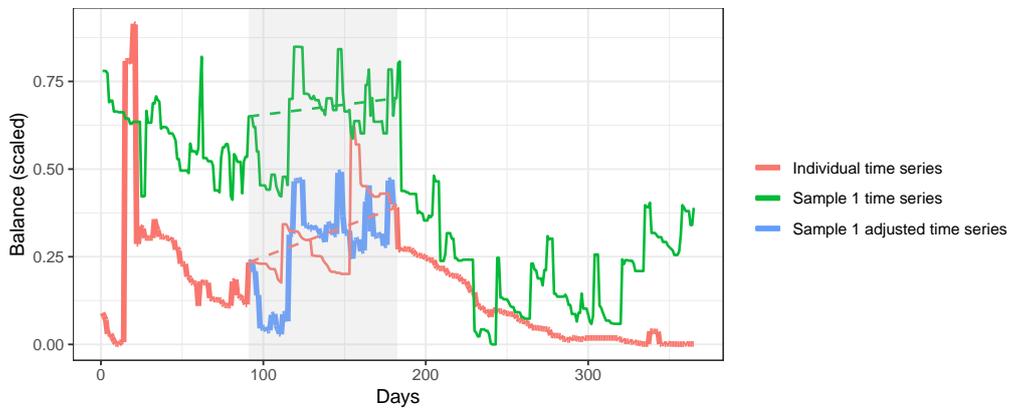


Figure 4: Time series of the individual we want to explain (solid red), one sample series (solid green) for period Q2. The straight dashed lines are fit to the end points of the two series. The adjusted time series (solid blue) is the green series adjusted to the dashed red line. This adjusted series is used to estimate $v(\{Q1, Q3, Q4\})$.

In order to put more weight on ‘similar’ sampled time series, the samples are weighted based on the Euclidean distance between the known parts of the samples and the individual series using a Gaussian kernel. Finally, $v(\{Q1, Q3, Q4\})$ is estimated for one individual by taking a weighted average of the predictions produced by these samples.

The *groupShapley* values of four fictive individuals are shown in Figure 5. The *groupShapley* values are small for individual 1. The last two quarters increase the probability of default, perhaps as a consequence of increased and reduced balance amounts, respectively, as seen in Figure 3. Individual 2 also has a small probability of default, and here Q2 stands out with the largest *groupShapley* value, perhaps as a consequence of rapid movements in the balance combined with an overall low balance. The balance increases significantly in Q4, which is a likely reason for its negative *groupShapley* value.

The last two individuals have rather high probabilities of default. For individual 3, one may guess that the high volatility in Q1 leads to the high probability of default. However, based on Figure 3, it is rather the constant decrease in the balance throughout Q3 and Q4 that increase

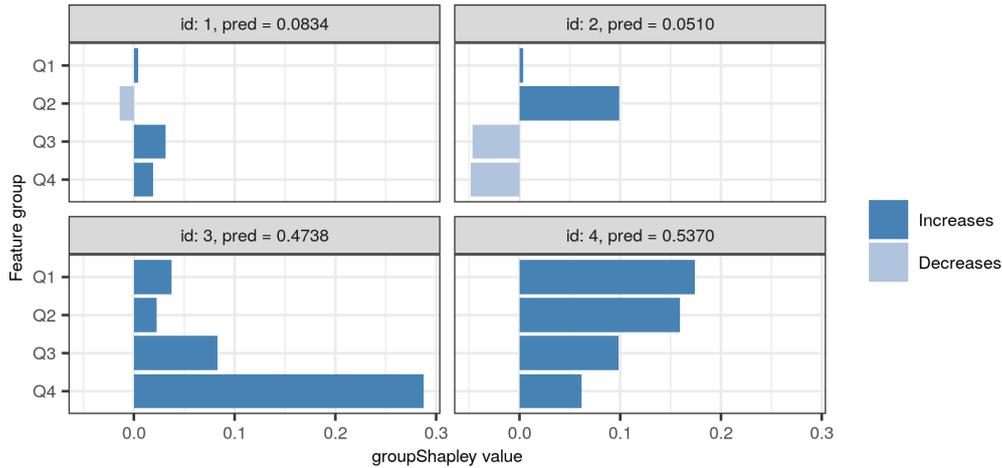


Figure 5: Estimated *groupShapley* values for four individuals in the time series default prediction example, using yearly quarters as groups.

the probability the most – which be naturally explained by the correspondence between lack of income and financial problems. For the fourth individual, all quarters increase the probability significantly, but Q1 and Q2 are the most significant. This is perhaps due to the very low account balance in these periods. Surprisingly, Q3 is not as decisive, but this might be because the balance increases steeply in this period.

This example illustrates the use of *groupShapley* on a time series classification problem. Time series problems are a natural use case for *groupShapley* because of the natural groupings of the features (time segments), and the number of features (time points) is almost always more than what is computationally feasible with Shapley values methodology.

5. Conclusion

We have introduced *groupShapley* and showcased how the method can be used to explain predictions from complex machine learning models. As long as the number of groups is limited, the approach bypasses the well-known computational bottleneck of feature-wise Shapley values. Through a series of real data use cases, we demonstrated how *groupShapley* provides efficient, simple, and informative prediction explanations.

In addition to the examples discussed in the paper, there is a range of other use cases where *groupShapley* is particularly relevant. One example is the explanation of individual categorical features which have been encoded (using e.g., one-hot encoding) as numeric features in order to use a predictive model that requires numeric input, see also [29]. This might be viewed as an alternative to the approach in [6] for computing feature-wise Shapley values in the presence of categorical features. Another example is in image classification. It is well known that deep learning models used in modern image classification are notoriously difficult to understand. By grouping single pixels into groups (or super pixels [45]), *groupShapley* can be used to explain such models.

Moreover, *groupShapley* is well suited to high dimensional health and mortality problems

in bio statistics and epidemiology. For example, *groupShapley* could be used with data from a national health directory (e.g., NHANES [46]) to explain groups of features like demographics, diet, lab results, and medications, rather than the hundreds of individual features typically present in such data sets. Finally, *groupShapley* is a completely general method that can be used in a range of other Shapley value settings. In addition to the prediction explanation setting, it can be used analogously as a *global* explanation method [47, 48], or even outside the fields of XAI and interpretable machine learning [49].

Acknowledgements

We thank Anders Løland for useful discussions in the early-stages of the paper. We also thank Jens Christian Wahl for support with the time series default prediction data and model. This work was supported by the Norwegian Research Council grant 237718 (Big Insight).

References

- [1] C. Molnar, Interpretable machine learning: A guide for making black box models explainable, Christoph Molnar, Leanpub, 2020.
- [2] L. S. Shapley, A Value for N-Person Games, Contributions to the Theory of Games 2 (1953) 307–317.
- [3] S. M. Lundberg, S.-I. Lee, A Unified Approach to Interpreting Model Predictions, in: Advances in Neural Information Processing Systems, Curram Associates Inc., 2017, pp. 4768–4777.
- [4] K. Aas, M. Jullum, A. Løland, Explaining individual predictions when features are dependent: More accurate approximations to shapley values, arXiv preprint arXiv:1903.10464 (2019).
- [5] H. Chen, J. D. Janizek, S. Lundberg, S.-I. Lee, True to the model or true to the data?, arXiv preprint arXiv:2006.16234 (2020).
- [6] A. Redelmeier, M. Jullum, K. Aas, Explaining predictive models with mixed features using shapley values and conditional inference trees, in: International Cross-Domain Conference for Machine Learning and Knowledge Extraction, Springer, 2020, pp. 117–137.
- [7] V. Conitzer, T. Sandholm, Computing shapley values, manipulating value division schemes, and checking core membership in multi-issue domains, in: AAAI, volume 4, 2004, pp. 219–225.
- [8] K. Corder, K. Decker, Shapley value approximation with divisive clustering, in: 2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA), IEEE, 2019, pp. 234–239.
- [9] S. S. Fatima, M. Wooldridge, N. R. Jennings, A linear approximation method for the shapley value, Artificial Intelligence 172 (2008) 1673–1699.
- [10] H. A. Soufiani, D. X. Charles, D. M. Chickering, D. C. Parkes, Approximating the shapley value via multi-issue decomposition, Proceedings of the International Foundation for Autonomous Agents and Multiagent Systems (2014).
- [11] G. Van den Broeck, A. Lykov, M. Schleich, D. Suci, On the tractability of shap explanations, in: Proceedings of AAAI, 2021.

- [12] J. Chen, L. Song, M. J. Wainwright, M. I. Jordan, L-shapley and c-shapley: Efficient model interpretation for structured data, in: International Conference on Learning Representations, 2019.
- [13] I. Covert, S.-I. Lee, Improving kernelshap: Practical shapley value estimation via linear regression, arXiv preprint arXiv:2012.01536 (2020).
- [14] X. Li, N. C. Dvornek, Y. Zhou, J. Zhuang, P. Ventola, J. S. Duncan, Efficient interpretation of deep learning models using graph structure and cooperative game theory: Application to asd biomarker discovery, in: International Conference on Information Processing in Medical Imaging, Springer, 2019, pp. 718–730.
- [15] S. M. Lundberg, S.-I. Lee, Consistent feature attribution for tree ensembles, in: Proceedings of the 34 th International Conference on Machine Learning, JMLR: W&CP, 2017, pp. 15–21.
- [16] K. Aas, T. Nagler, M. Jullum, A. Løland, Explaining predictive models using shapley values and non-parametric vine copulas, Dependence Modeling 9 (2021) 62–81.
- [17] M. Jullum, A. Redelmeier, K. Aas, groupshapley: Efficient prediction explanation with shapley values for feature groups, arXiv preprint arXiv:2106.12228 (2021).
- [18] N. Sellereite, M. Jullum, shapr: An r-package for explaining machine learning models with dependence-aware shapley values, Journal of Open Source Software 5 (2020) 2027.
- [19] R. J. Aumann, J. H. Dreze, Cooperative games with coalition structures, International Journal of game theory 3 (1974) 217–237.
- [20] Y. Kamijo, The collective value: a new solution for games with coalition structures, Top 21 (2013) 572–589.
- [21] G. Owen, Values of games with a priori unions, in: Mathematical economics and game theory, Springer, 1977, pp. 76–88.
- [22] J. Vidal-Puga, The harsanyi paradox and the “right to talk” in bargaining among coalitions, Mathematical Social Sciences 64 (2012) 214–224.
- [23] M. Grabisch, M. Roubens, An axiomatic approach to the concept of interaction among players in cooperative games, International Journal of game theory 28 (1999) 547–565.
- [24] G. Owen, Multilinear extensions of games, Management Science 18 (1972) 64–79.
- [25] R. Flores, E. Molina, J. Tejada, The shapley group value, arXiv preprint arXiv:1412.5429 (2014).
- [26] R. Flores, E. Molina, J. Tejada, Evaluating groups with the generalized shapley value, 4OR 17 (2019) 141–172.
- [27] J.-L. Marichal, I. Kojadinovic, K. Fujimoto, Axiomatic characterizations of generalized values, Discrete Applied Mathematics 155 (2007) 26–43.
- [28] S. Jeong, Y. Shoham, Marginal contribution nets: A compact representation scheme for coalitional games, in: Proceedings of the 6th ACM Conference on Electronic Commerce, 2005, pp. 193–202.
- [29] S. I. Amoukou, N. J. Brunel, T. Salaün, Accurate and robust shapley values for explaining predictions and focusing on local important variables, arXiv preprint arXiv:2106.03820 (2021).
- [30] Q. Au, J. Herbringer, C. Stachl, B. Bischl, G. Casalicchio, Grouped feature importance and combined features effect plot, arXiv preprint arXiv:2104.11688 (2021).
- [31] A. Miroshnikov, K. Kotsiopoulos, A. R. Kannan, Mutual information-based group explainers with coalition structure for machine learning model explanations, arXiv preprint

arXiv:2102.10878 (2021).

- [32] A. Shanbhag, A. Ghosh, J. Rubin, Unified shapley framework to explain prediction drift, arXiv preprint arXiv:2102.07862 (2021).
- [33] L. Kaufman, P. J. Rousseeuw, Finding groups in data: an introduction to cluster analysis, volume 344, John Wiley & Sons, 2009.
- [34] C. Frye, D. de Mijolla, L. Cowton, M. Stanley, I. Feige, Shapley-based explainability on the data manifold, arXiv preprint arXiv:2006.01272 (2020).
- [35] T. Fawcett, An introduction to roc analysis, Pattern recognition letters 27 (2006) 861–874.
- [36] G. W. Brier, et al., Verification of forecasts expressed in terms of probability, Monthly weather review 78 (1950) 1–3.
- [37] M. N. Wright, A. Ziegler, ranger: A fast implementation of random forests for high dimensional data in C++ and R, Journal of Statistical Software 77 (2017) 1–17. doi:10.18637/jss.v077.i01.
- [38] M. E. Burczynski, R. L. Peterson, N. C. Twine, K. A. Zuberek, B. J. Brodeur, L. Casciotti, V. Maganti, P. S. Reddy, A. Strahs, F. Immermann, et al., Molecular classification of crohn’s disease and ulcerative colitis patients using transcriptional profiles in peripheral blood mononuclear cells, The journal of molecular diagnostics 8 (2006) 51–61.
- [39] N. Simon, J. Friedman, T. Hastie, R. Tibshirani, A sparse-group lasso, Journal of computational and graphical statistics 22 (2013) 231–245.
- [40] A. Liberzon, C. Birger, H. Thorvaldsdóttir, M. Ghandi, J. P. Mesirov, P. Tamayo, The molecular signatures database hallmark gene set collection, Cell systems 1 (2015) 417–425.
- [41] R. Tibshirani, Regression shrinkage and selection via the lasso, Journal of the Royal Statistical Society: Series B (Methodological) 58 (1996) 267–288.
- [42] X. Lu, Y. Yu, S. Tan, p53 expression in patients with ulcerative colitis-associated with dysplasia and carcinoma: a systematic meta-analysis, BMC gastroenterology 17 (2017) 1–8.
- [43] M. B. Braga-Neto, J. M. Gaballa, A. O. Bamidele, O. F. Sarmiento, P. Svingen, M. Gonzalez, G. P. Ramos, M. R. Sagstetter, S. O. Aseem, Z. Sun, et al., Dereglulation of long intergenic non-coding rnas in cd4+ t cells of lamina propria in crohn’s disease through transcriptome profiling, Journal of Crohn’s and Colitis 14 (2020) 96–109.
- [44] H. Kvamme, N. Sellereite, K. Aas, S. Sjursen, Predicting mortgage default using convolutional neural networks, Expert Systems with Applications 102 (2018) 207–217.
- [45] M. T. Ribeiro, S. Singh, C. Guestrin, Why should I trust you? Explaining predictions of any classifier, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), ACM, 2016, pp. 1135–1144.
- [46] E. E. Hatch, J. W. Nelson, M. M. Qureshi, J. Weinberg, L. L. Moore, M. Singer, T. F. Webster, Association of urinary phthalate metabolite concentrations with body mass index and waist circumference: a cross-sectional study of nhanes data, 1999–2002, Environmental Health 7 (2008) 1–15.
- [47] U. Grömping, Estimators of relative importance in linear regression based on variance decomposition, The American Statistician 61 (2007) 139–147.
- [48] A. B. Owen, C. Priour, On Shapley value for measuring importance of dependent inputs, SIAM/ASA Journal on Uncertainty Quantification 5 (2017) 986–1002.
- [49] S. Moretti, F. Patrone, Transversality of the shapley value, Top 16 (2008) 1–41.