Efficient Shapley value explanation through feature groups

Martin Jullum[1], **Annabelle Redelmeier**[2] **and Kjersti Aas**[3]

[1] *Norwegian Computing Center, Norway, jullum@nr.no*
[2] *Norwegian Computing Center, Norway, anr@nr.no*
[3] *Norwegian Computing Center, Norway, kjersti@nr.no*

Shapley values has established itself as one of the most appropriate and theoretically sound frameworks for explaining predictions from complex regression/machine learning models. The popularity of Shapley values in the explanation setting is probably due to its unique theoretical properties. The main drawback with Shapley values, however, is that its computational complexity grows exponentially in the number of input features (covariates), 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 becomes challenging, see e.g. [1].

I hereby present *groupShapley*: a conceptually simple approach for dealing with the aforementioned bottlenecks. The idea is to group the features, for example by type or dependence, 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, makes precise computations practically feasible and the presentation and knowledge extraction greatly simplified.

It may be shown that under certain conditions, groupShapley is equivalent to summing the feature-wise Shapley values within each feature group, but simulations indicate that they may differ significantly outside the conditions. The groupShapley method is implemented in a development version of the R-package `shapr` [2].

I will introduce the method, and talk about it from both theoretical and practical sides. The practical simplifications will be showcased through a real world car insurance example.

References

[1] Aas, K., Jullum, M., Løland, A (2021). Explaining individual predictions when features are dependent: More accurate approximations to Shapley values. *Artificial Intelligence*, **298**, https://doi.org/10.1016/j.artint.2021.103502

[2] Sellereite, N. and Jullum, M. (2020). shapr: An R-package for explaining machine learning models with dependence-aware Shapley values. *Journal of Open Source Software*, **5**(46), 2027, https://doi.org/10.21105/joss.02027

Why Explainable AI Should Move from Influence to Contextual Importance and Utility

Kary Främling

Department of Computing Science, Umeå University

Contextual Importance and Utility (CIU) is a method originally developed by Kary Främling in his PhD thesis [1]. CIU's objective was to enable similar explanation concepts for all possible decision support models, ranging from linear models such as the weighted sum, to rule-based systems, decision trees, fuzzy systems, neural networks and any machine learning-based models.

CIU arithmetic is defined using utility functions, which are a cornerstone in Decision Theory and notably in Multi-Attribute Utility Theory. Both Contextual Importance (CI) and Contextual Utility (CU) values are in the range $[0, 1]$ and have absolute (i.e. not relative) interpretations. $CI = 0$ signifies that the studied input feature (or set of features i) cannot change the output value (or its utility, in practice), no matter how much the value of the input feature changes. $CI = 1$ again signifies that the output value can change totally within the possible value range. $CU = 0$ signifies that the current value of the input feature(s) is the worst possible for the output value, whereas $CU = 1$ signifies that it's the best possible value.

Most current outcome explanation methods belong to the family of additive feature attribution methods, which in practice produce what we call an *influence* value ϕ. Contrary to CI and CU, ϕ is a relative value that is more difficult to define and interpret. Furthermore, it can be shown that influence only is incapable of producing any explanation even in simple settings. The presentation shows why CIU should be the method of choice for producing model-agnostic explanations of black-box outcomes. CIU is also defined for so-called intermediate concepts, which makes it possible to structure explanations in a hierarchical way that influence-based methods might not be able to provide.

References

[1] Främling, K.: Modélisation et apprentissage des préférences par réseaux de neurones pour l'aide à la décision multicritère. PhD thesis, INSA de Lyon (1996)

Model interpretability in Bayesian framework

Homayun Afrabandpey

Nokia Technologies, Finland, homayun.afrabandpey@nokia.come

A salient approach to obtain interpretability in the Bayesian framework is to restrict the prior to favor interpretable models. This approach, however has shortcomings including difficulties in formulating interpretability prior for certain classes of models such as neural networks and obtaining good trade-offs between accuracy and interpretability. In this talk, I will explain our recent approach towards interpreting Bayesian predictive models through a decision theoretic approach without constraining priors. We developed a two-step strategy to interpretability in the Bayesian framework. In the first step, we fit a highly accurate Bayesian predictive model, which we call reference model, to the training data without constraining it to be interpretable. In the second phase, we construct an interpretable proxy model that best describes locally and/or globally the behavior of the reference model. In our experiments, we show that for the same level of interpretability, our approach finds better trade-off and generates more accurate models than the earlier alternative of restricting the prior.

This is a joint work with Tomi Peltola, Juho Piironen, Aki Vehtari and Samuel Kaski, available in https://link.springer.com/article/10.1007/s10994-020-05901-8.