Shapley value for feature evaluation in a mushroom edibility classification problem

Francesco Lorenzi, Giacomo De Lazzari
Department of Information Engineering, University of Padova,
email: {francesco.lorenzi.2, giacomo.delazzari.1}@studenti.unipd.it

***Abstract*—Starting from the previous applications of Shapley values to Machine Learning (ML) problems of explanation, pruning, and evaluation, we apply the technique in the peculiar case of a dataset for mushroom edibility classification. The samples in the dataset are characterized by qualitative features like the ones used by mushrooms pickers. The feature space presents interesting aspects which are discussed and experimented.**

I. Introduction

The Shapley value is a cooperative game theory concept which gained popularity in the ML fields of model interpretation and explanation. This concept can be applied in the evaluation procedure of samples in training sets, as well as features in the feature spaces, and models in model ensembles. Its flexibility stems from the definition in original works from cooperative game theory, for which it is only required to model the desired evaluation scenario in terms of players cooperating in coalitions, with given outcomes. Being that this topic crosses the border between ML and cooperative game theory, in the next paragraph we connect the existing literature and results.

A. Previous work

The concept of the Shapley value appeared for the first time in the original work by Shapley [10], in 1952, as a *value* for cooperative game theory. In this work, the derivation of the value starts from a set of axioms that are of interest in the cooperative theory setting. In recent times the definition of the value has been generalized even to groups of players, as explained in Flores et al. [5]. In a parallel track with respect to cooperative game theory, the ML field developed solutions to crucial problems of feature and model evaluation and selection. An example is shown in Peng et al. [9] in which criteria called *maximal statistical dependency* and *minimal-redundancy-maximal-relevance (mRMR)* are derived from information theory arguments.

The connection between the two fields is explained in the case of classifier model evaluation in Rozemberczki and Sarkar [2], and for data evaluation in Ghorbani and Zou [6]. The feature selection application of the value is thoroughly discussed in Sun et al. [11].

Shapley value can be hard to compute as the scale of the problem increases, so a variety of approximation algorithms has been developed. Some well-performing techniques are described in Ghorbani and Zou [6], and also in Castro et al. [3]. Many implementations of Shapley value analysis in ML adopt a framework called SHAP (SHapley Additive exPlanations) presented in [8], that aims to *explain* models.

An example of the effectiveness of the Shapley value in ML is shown in Alsuradi et al. [1] where EEG data classification was analyzed with SHAP in order to highlight important features.

B. Paper organization

The various sections are organized as follows. In section II we start from the original work of [10], then we build on this with the aim of reaching a formulation applicable to machine learning problems. We consider the previous work in [2] and [6], in which the framework of application of Shapley value in ML is explained for the cases, respectively, of model and data evaluation. Then we comment the issues of applying such theory to an ML problem. In second place we comment on the adoption of the same Monte Carlo algorithm as in [6] for the practical computation of the value over the selected dataset of mushrooms from UCI [4].

In section III we report the result of the computation and interpret it with respect to the properties of the mushrooms.

Two appendices are given to shed light on some theoretical aspects regarding cooperative game theory (Appendix A), and the probabilistic expression of Shapley value (Appendix B). An additional part (Appendix C) is devoted to proving a property of a specific ML technique (*one-hot encoding*) which was useful for the ML model selection.

The paper aims to show a possible implementation of the Shapley value technique for feature evaluation, highlighting the most important theoretical and algorithmic difficulties, and to show the interpretative potential of this technique when applied to a dataset designed for mushroom edibility classification.

II. Theoretical discussion

A. Preliminary: cooperative game theory

The original paper from Shapley [10] gives the definition of Shapley values in a precise game-theoretic scenario: the one of cooperative theory, in which a *characteristic function* of the game is defined. Instead of focusing on individual behavior, by using the concepts of best responses and Nash equilibria, as done in non-cooperative game theory, the point of view of the cooperative theory, for what concerns our work, is that a utility function is assigned to a group of players, as a "team". Let us describe some details of this framework. If the set of

all possible players is D, and $\mathcal{P}(D)$ indicates the power set of D, the *characteristic function* of the game (sometimes called "the game" itself by synecdoche) is

$$v : \mathcal{P}(D) \longrightarrow \mathbb{R} \tag{1}$$

This function, which we call v, is usually required to satisfy the following properties

$$v(\emptyset) = 0 \tag{2}$$

$$v(S) > v(S \cap T) + v(S - T) \tag{3}$$

where the last property is called the *superadditivity* property. The characteristic function $v(S)$ represents a collective payoff for the collaborating group S. With this framework we can define the concept of *value*. The value is a function

$$\phi : D \longrightarrow \mathbb{R} \tag{4}$$

which associates every player with a real number that represents the payoff of a player when it is cooperating in the group D. In Shapley [10] the value is required to satisfy a set of axioms [10, axioms 1-2-3] regarding its algebraic properties. It can be proven that the only function ϕ which satisfies the set of axioms is the one defined as

$$\phi_i(v) = \sum_{S \subseteq D - \{i\}} \frac{1}{n} \binom{n-1}{|S|}^{-1} [v(S \cup \{i\}) - v(S)] \tag{5}$$

being $n = |D|$. Notice that the function depends on the choice of the game, so $\phi = \phi(v)$. This function assigns to every player its *Shapley value* $\phi_i(v)$. Such function also satisfies the so-called *efficiency* property

$$\sum_{i \in D} \phi_i(v) = v(D). \tag{6}$$

In order to apply Shapley value theory in an ML setting, we need to set up some definitions that allow the interpretation of *features* as *players* in a cooperative game. These definitions can mimic the approach in [2, sec. 3] in which the central idea is the one of *ensemble game*. We can use the same ideas even though the focus of that work is towards the evaluation of models. One tricky and crucial quantity to define is the analogue of the *characteristic function* for an ML learning process. Since our focus is the evaluation of features, given a subset $S \subseteq D$ of features to consider, we train a model of the chosen class using a fixed training set of samples of which we only consider the features in S. We then compute the empirical score on a fixed test set, also considering the same subset of features. The natural candidate for the characteristic function then is the *score* function over a fixed test set.

However, we know from ML that, in the case of binary classification, which is our interest, the worst obtainable score is 0.5, and not 0. The 0 score corresponds to a classifier that always classifies wrong, so in a sense it is still able to perfectly classify the data, just as one with score 1. We wish to define a characteristic function which is as similar as possible to the one used in cooperative theory, so that we can use its properties. In order to satisfy the property (2), we choose as characteristic function the following one

$$v(S) = 2T(S) - 1 \tag{7}$$

where $T(S)$ is the empirical score computed as described before.

B. On super- and sub-additivity

The remaining axiom (3) requires some additional comments. In an ML setting, it may be possible that the performance of a model decreases as the set of features becomes more numerous. This is not surprising at all, since we are allowing for a general model class, and the model's algorithm may respond in unexpected ways to different feature sets. So if we adopt the score function as in (7) as the game function, it may not satisfy property (3) for a generic algorithm. Apparently there is no easy way to obtain superadditivity in a ML scenario. In absence of such property, let us comment on a key passage: the role of a superadditive characteristic function, from a non-cooperative theory point of view.

Suppose to have a static game of complete information in which there are n players. Every player has the choice of C cooperate or N not cooperate. The outcome of the game is a set of collaborating players called "the coalition" (who collaborate, together), and a set of not collaborating players (who play for themselves, individually). Every player i gets a payoff which is its individual payoff $v(\{i\})$ if it is not collaborating, or its Shapley value $\phi_i(v, S)$ (where S indicates the coalition formed) if it is collaborating. It is possible to prove that a Nash equilibrium in presence of superadditive characteristic function is $(C, C, ..., C)$ (they all cooperate) because of the following property:

$$\phi_i(v, S) \geq v(\{i\}) \qquad \forall S, i \tag{8}$$

This simple argument connects this aspect of cooperative game theory with non-cooperative game theory. A proof of what stated is addressed in Appendix A.

Vice versa, in a scenario in which the characteristic function is not superadditive, the player strategies may include the possibility not to cooperate, even as part of a mixed strategy. In this way the whole argument of Shapley values, which is computed with respect to the coalition of all players, would not make sense from the individual point of view. The solution of this apparent dilemma, in the application of Shapley values to ML, is done by the following observation: if the players do not have the choice to cooperate or not, but they are only passive entities, the usage of non-superadditive v is justified, as they may be "forced" to cooperate even if whis implies a reduction in their individual utility. In fact, in the case of ML the features are not proper players, as they are selected a priori, and the forcing agent makes the choice of the group composition. This is the point of view adopted in [6] for the problem of evaluation of data. In conclusion, from the point of view of ML, a non-superadditive function gives a consistent generalization of Shapley values.

However, in the following sections, we notice that for a sufficiently well-performing learning algorithm, the score behaves very often as a superadditive function.

C. Characteristic function as an ensemble average

The computation of the Shapley values for a large number of features is computationally expensive. However, a different expression can be written for $\phi_i(v)$, as originally suggested in [3],

$$\phi_i(v) = \mathbb{E}_{\pi\sim\Pi}\left[v(S^i_\pi \cup \{i\}) - v(S^i_\pi)\right] \tag{9}$$

where Π is a uniform distribution over the permutations of the features, and S^i_π is the set of all features which, in the order of the permutation, come before feature i. This alternative expression for ϕ is useful as it leans towards the usage of a Monte Carlo technique to estimate the value of the function.

The step-by-step derivation of this expression from (5) is given in Appendix B.

D. Dataset and model

Fig. 1. Example of a mushroom from *Lepiota* genus

Let us consider the actual dataset of choice [4]. It is a set of real data extracted by observing some mushrooms from the genus *Lepiota*, similar to the one in Figure 1. Some of the species included are edible, some others are poisonous. Every feature takes value in a discrete class of categories, as shown in Table I. Those values represent the mushroom characteristics, typically recommended by mushrooms manuals, which are useful to mushroom pickers as they are readily observable and interpretable.

We have chosen to model the classifier with a decision tree. At first, we have been experimenting with a linear classifier since we initially believed it would be one of the simplest machine learning models. Moreover, we believed it to be able to draw satisfying enough decision boundaries *given that the features get one-hot encoded*. In fact, all of the features are *categorical*, as is often stated in the machine learning field. For a linear model, it is not reasonable to represent them as real values since there is (for instance) no order relation in their domain. Take as an example the *cap-surface* feature: it can belong to the categories {*fibrous*, *grooves*, *scaly*, *smooth*} as per dataset description [4]. Then *one-hot encoding* is a natural representation, for which every feature is mapped to a "one-hot vector" whose dimension is equal to the number of categories. However, we understood that such encoding, when applied to all the features to obtain a numerical feature vector given by the concatenation of all the single one-hot encoded vectors, doesn't guarantee linear separability for arbitrary labelings. A proof of this fact is given in Appendix C.

i	name	λ_i
1	*cap-shape*	6
2	*cap-surface*	4
3	*cap-color*	10
4	*bruises*	2
5	*odor*	9
6	*gill-attachment*	4
7	*gill-spacing*	3
8	*gill-size*	2
9	*gill-color*	12
10	*stalk-shape*	2
11	*stalk-root*	6
12	*stalk-surface-above-ring*	4
13	*stalk-surface-below-ring*	4
14	*stalk-color-above-ring*	9
15	*stalk-color-below-ring*	9
16	*veil-type*	2
17	*veil-color*	4
18	*ring-number*	3
19	*ring-type*	8
20	*spore-print-color*	9
21	*population*	6
22	*habitat*	7

TABLE I
FEATURES OF THE DATASET

A decision tree was ultimately used, since it is instead able to encode any labeling. Additionally, the learning procedure is computationally faster than the (gradient descent based) one used for a linear classifier, and this is quite relevant given our technique of choice for the estimation of the Shapley values.

E. Computation of Shapley values

Starting from equation (9), we can adopt the efficient Monte Carlo algorithm proposed in [6, algorithm 1]. The algorithm is called Truncated Monte Carlo (TMC), and the truncation point is set following empirical indications in [6]. In particular, the sampling set is selected to be larger than $3n$, where n is the number of features, in order to have a sufficient convergence.

The TMC algorithm has been implemented in the Python programming language, exploiting the popular NumPy and scikit-learn packages. A method `tmc_shapley` is used to perform the estimation, provided various parameters such as the number of samples, the train and test dataset, the classifier to use, and the characteristic function v as in [7]. As for the decision tree model, the one provided by scikit-learn is used, then $v(S)$ is computed by evaluating the resulting trained model on the test dataset. To improve the execution time of the TMC algorithm, the early termination condition discussed in [6] has also been implemented.

The implementation is available on GitHub [7], where the development history is accessible. As a consequence, all of the experiments regarding model choice, with the initial evaluation of a linear classifier, can be seen.

III. Experimental results

The TMC algorithm has been used to produce the results depicted in Figure 2. The order of magnitude of the number

of samples has been set around the 10^5 mark.

First off, we validated that by computing the sum of all the estimated Shapley values, the resulting quantity is indeed 1 (more specifically, in our simulation, it is experimentally off only by the fraction $4 \cdot 10^{-16}$). This is coherent with the properties of the value.

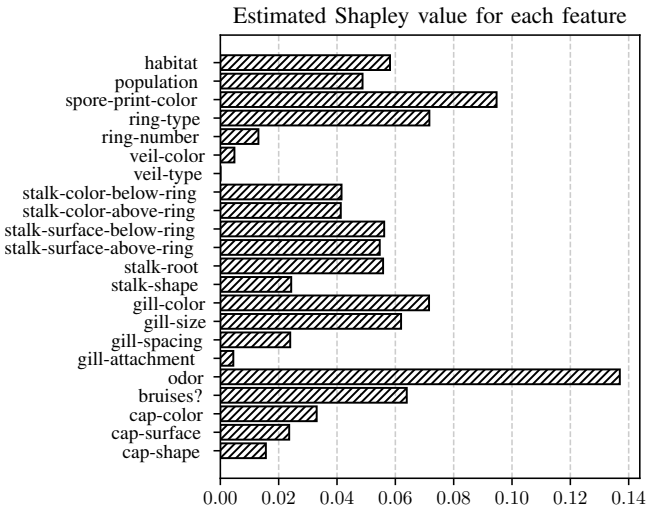

Fig. 2. Estimated Shapley values of the *feature players*

We are now provided with a graphical representation that is useful to interpret the model. By looking at the chart we might infer that the most important mushroom aspect to watch out for is its odor, followed by its spores color. Some other aspects (such as the ring type, gill size and color, whether the mushroom has bruises or not, etc) play a moderate role. On the other hand, we can see that looking at the veil and the attachment of the gill is not very useful, together with the number of rings which doesn't seem to be particularly important in order to make a good prediction regarding the edibility.

These results might also be used to understand how much identification accuracy one can expect if able to clearly identify only a subset of the features. It is worth noting, though, that such interpretation would formally require the evaluation of groups of features, which is a topic that has not been explored in our work.

Anyhow, we argue that these experimental results can be very useful to an amateur mushroom picker as an advise on which mushroom attributes deserve higher attention.

IV. Conclusions

In conclusion, we believe that the work we presented shows a successful application of the Shapley value, as a tool from cooperative game theory, to a machine learning setting. The initial objective of obtaining interpretative results related to the ML problem of mushroom classification has been achieved, with the understanding of the most relevant features of the mushroom relative to the accuracy of identifying an edible specimen. The goal we set in the introduction of obtaining a formulation based on Shapley values for feature evaluation has been fulfilled, with the experimental results validating the properties we highlighted in the theoretical part.

At the end of the previous section, we also touched upon the possibility of exploring the evaluation of groups of features. We establish this to be a possible continuation of the work presented in this paper. The generalized Shapley value introduced in Flores et al. [5] in the context of cooperative game theory looks like a valid starting point for the theoretical aspects of a hypothetical adaptation to the evaluation of feature groups.

Appendix

Appendix A - Solution of the example game

We provide in this appendix a simple solution of the game described in II-B. First of all, let us introduce some notation belonging to cooperative game theory, and derive some known result in our notation. Let D be the set of all players, as defined before.

Definition 1. *An **imputation** is a solution ϕ of a cooperative game with characteristic function v in which it holds:*

$$\phi_i(v, S) \geq v(\{i\}) \qquad \forall S, i \in S \tag{10}$$

Theorem 1. *If v is superadditive, Shapley value solution $\phi(v, S)$ is an imputation for each outcome coalition S*

Proof. Recall that, by superadditivity, $\forall S, i$, it holds $v(S \cup \{i\}) - v(S) \geq v(\{i\})$. So we can write the inequality

$$\phi_i(v, S) = \tag{11}$$

$$= \sum_{S \subseteq D - \{i\}} \frac{1}{n} \binom{n-1}{|S|}^{-1} [v(S \cup \{i\}) - v(S)] \geq \tag{12}$$

$$\geq \sum_{S \subseteq D - \{i\}} \frac{1}{n} \binom{n-1}{|S|}^{-1} v(\{i\}) = \tag{13}$$

$$= \frac{v(\{i\})}{n} \sum_{S \subseteq D - \{i\}} \binom{n-1}{|S|}^{-1}. \tag{14}$$

The number of subsets of cardinality $|S|$ into a set of cardinality $n-1$ is

$$\binom{n-1}{|S|} \tag{15}$$

so we can rewrite the sum term summing over subsets cardinalities

$$\phi_i(v, S) \geq \frac{v(\{i\})}{n} \sum_{|S|=0}^{n-1} 1 = v(\{i\}). \tag{16}$$

□

At this point, let us analyze the proposed game in the framework of non-cooperative game theory. The outcome s of this game is described by the set S of players which collaborate together, which is represented by a n-uple of values $s = (s_1, \ldots, s_{|D|})$ which can be C or N for each player,

respectively meaning *collaborating* and *not collaborating*. The utility $u_i(s)$ for each player i will be

$$u_i(s) = \begin{cases} v(\{i\}) & \text{if } s_i = N \\ \phi_i(v, S) & \text{if } s_i = C \end{cases}. \tag{17}$$

The dependence of Shapley values on the other players choice is encoded in S itself. Using Theorem 1 we conclude that $s_i = C$ is a *dominant* strategy for each i, as it holds

$$u_i(C, s_{-i}) = \phi_i(v, S) \geq v(\{i\}) = u_i(N, s_{-i}). \tag{18}$$

Since every player i has a dominant strategy $s_i^* = C$ then $(C, C, ..., C)$ is a Nash equilibrium, as reported in II.

Appendix B - TMC: probabilistic expression

In II-C we obtain a probabilistic expression for the Shapley value. This is obtained by recognizing in the summation term an expectation summation with respect to a given distribution. Let us start from the original definition of the Shapley value in equation (5), then manipulate such expression

$$\phi_i(v) = \sum_{S \subseteq D - \{i\}} \frac{1}{n} \binom{n-1}{|S|}^{-1} [v(S \cup \{i\}) - v(S)] \tag{19}$$

$$= \sum_{S \subseteq D - \{i\}} \frac{1}{n!} |S|!(n - |S| - 1)! \, [v(S \cup \{i\}) - v(S)] \tag{20}$$

and notice that we can develop a different way to sum over $\mathcal{P}(D)$. Consider Π, an uniform distribution over the set of all the permutation of the set D, denoted by $\text{Perm}(D)$, and $\pi \sim \Pi$. Define S_π^i the *predecessor* of i with respect to the permutation π, which is

$$S_\pi^i = \{\pi(1), \pi(2), ..., \pi(k-1)\} \text{ where } \pi(k) = i. \tag{21}$$

For a fixed i, let us consider a fixed S in the sum of equation (20), and understand for which permutations π we have that $S_\pi^i = S$. We find that all the permutations we are looking for are such that $\pi(|S| + 1) = i$, and all the elements before $\pi(|S| + 1)$ are the elements of S. Considering all the possible permutations of S and the ones over the remaining $n - |S| - 1$ elements after $\pi(|S|+1)$, we have a set of allowable permutations $R(S)$ with cardinality:

$$|R(S)| = |S|!(n - |S| - 1)! \tag{22}$$

and this is exactly the coefficient we find in equation (20), so we can write

$$\phi_i(v) = \sum_{S \subseteq D - \{i\}} \frac{1}{n!} \sum_{\pi \in R(S)} \left[v(S_\pi^i \cup \{i\}) - v(S_\pi^i) \right] \tag{23}$$

to then conclude

$$\phi_i(v) = \sum_{\pi \in \text{Perm}(D)} \frac{1}{n!} \left[v(S_\pi^i \cup \{i\}) - v(S_\pi^i) \right] \tag{24}$$

which is precisely an expectation with respect to an uniform variable. Notice in fact that

$$|\text{Perm}(D)| = n! \tag{25}$$

so we obtain the equation (9)

$$\phi_i(v) = \mathbb{E}_{\pi \sim \Pi} \left[v(S_\pi^i \cup \{i\}) - v(S_\pi^i) \right]. \tag{26}$$

In Ghorbani and Zou [6] the above computation is done differently, as the free C coefficient of a generalized Shapley value written as

$$\widehat{\phi}_i(v) = C \sum_{S \subseteq D - \{i\}} \binom{n-1}{|S|}^{-1} [v(S \cup \{i\}) - v(S)] \tag{27}$$

is set to $1/n!$. We argue that this is not correct, as the sum is, referring to [3], a sum over permutations, not a sum over the power set. The constant $1/n!$ then arises naturally with the original definition of Shapley value, which is the one with $C = 1/n$. This is not only a mere issue of definitions and terminology since, for instance, such $\widehat{\phi}_i(v)$ do not satisfy the efficiency property (6) which is crucial in order to interpret the value as a game solution.

Appendix C - Linear non-separability of one-hot encoded feature vectors

While choosing the machine learning model we ended up wondering if one-hot encoding the features would lead to a linearly separable classification problem. We didn't easily find any proof or statement in the literature regarding this, so we explored and proved this result by ourselves.

Let us define, in our specific case of binary labeling and categorical-only features, a dataset sample as an object in

$$\mathcal{X} \times \mathcal{Y} = (\mathcal{X}_1 \times \mathcal{X}_2 \times \ldots \times \mathcal{X}_m) \times \{-1, 1\} \tag{28}$$

where the sets $\mathcal{X}_i$ contain the categories the i-th feature can belong to. We define $\lambda_i \coloneqq |\mathcal{X}_i|$ which represents the number of different categories the i-th feature can belong to. W.l.o.g. we assume $\mathcal{X}_i = \{x_i \in \mathbb{N} \mid 1 \leq x_i \leq \lambda_i\}$ i.e. that each category is a set of integer values.

A labeling is a map $\mathcal{X}_1 \times \mathcal{X}_2 \times \ldots \times \mathcal{X}_m \to \mathcal{Y}$ that assigns a binary label to any possible combination of categories each feature belongs to.

One-hot encoding of the i-th feature can then be defined as the one-to-one mapping

$$\begin{aligned} \omega_i \colon \mathcal{X}_i &\to O_i \\ x_i &\mapsto \vec{e}_{x_i} \end{aligned} \tag{29}$$

where $\vec{e}_j$ is the j-th canonical vector of $\mathbb{R}^{\lambda_i}$, so O_i is the set of allowed one-hot encoded λ_i-dimensional vectors for the i-th feature and is defined as

$$O_i = \{\vec{e}_j \in \mathbb{R}^{\lambda_i} \mid j \in [1, \lambda_i] \subset \mathbb{N}\} \tag{30}$$

(i.e. $o_i \coloneqq \omega_i(x_i) \in O_i$ is such that only one of its components is set to 1, with the others being 0).

A sample with multiple features is encoded as a feature vector obtained by concatenating the $\omega_i(x_i)$ of all the features. Thus we define the one-hot encoder for a sample as

$$\Omega \colon \mathcal{X} \to O \tag{31}$$

$$(x_1, \ldots, x_m) \mapsto (\omega_1(x_1), \ldots, \omega_m(x_m)) \tag{32}$$

Where $O \coloneqq \bigtimes_{i=1}^{m} O_i \subset \mathbb{R}^n$ and $n \coloneqq \sum_{i=1}^{m} \lambda_i$. We prove that for $m \geq 2$ there exists at least a labeling function on O which is not linearly separable[1].

Theorem 2. *With the notation introduced above, let $\mathcal{B} = \{\beta \colon X \to \{-1, 1\}\}$ be the set of binary labelings of objects identified by m categorical features.*

Then $m \geq 2 \implies \exists \beta \in \mathcal{B} \mid \nexists \ell \in \mathcal{L}$ such that $\ell \circ \Omega = \beta$, where $\mathcal{L} = \{\ell \colon \mathbb{R}^n \to \{-1, 1\}\}$ is the set of linear classifiers that map $z \mapsto \text{sign}(z^T w + b)$.

Proof. Assume $m \geq 2$ and $\lambda_i \geq 2 \ \forall i = 1, \ldots, m$ so that we have a meaningful structure. Let us consider a special $\bar{\beta} \in \mathcal{B}$ with the following property:

$$\bar{\beta}(x_1, x_2, \ldots, x_m) = \begin{cases} \bar{\alpha}(x_1) & \text{if } x_2 = 1 \\ -\bar{\alpha}(x_1) & \text{if } x_2 \neq 1 \end{cases} \tag{33}$$

where $\bar{\alpha} \colon \mathcal{X}_1 \to \{-1, 1\}$ is surjective, that is

$$\exists \ s, t \in \mathcal{X}_1 \mid \bar{\alpha}(s) = 1, \ \bar{\alpha}(t) = -1, \ s \neq t \tag{34}$$

So $\bar{\beta}$ depends only on the first two features x_1 and x_2 and exploits a binary labeling $\bar{\alpha}$ which only depends on the first feature x_1, negating its output based on the second feature x_2. Let us prove that such $\bar{\beta}$ is not linearly separable by contradiction. Suppose that $\exists \ell \in \mathcal{L}$ such that $\ell \circ \Omega = \bar{\beta}$. Since ℓ is a linear classifier, there must be some $w \in \mathbb{R}^{\lambda_1}, \ v \in \mathbb{R}^{\lambda_2}, \ b \in \mathbb{R}$ such that

$$\text{sign}(\omega_1(x_1)^T w + \omega_2(x_2)^T v + b) = \begin{cases} \bar{\alpha}(x_1) & \text{if } x_2 = 1 \\ -\bar{\alpha}(x_1) & \text{if } x_2 \neq 1 \end{cases} \tag{35}$$

for all x_1, x_2 and specifically for $x_1 \in \{s, t\}$ with s, t as in (34). So for $x_1 = s$ we obtain

$$\text{sign}(\omega_1(s)^T w + \omega_2(x_2)^T v + b) = \begin{cases} 1 & \text{if } x_2 = 1 \\ -1 & \text{if } x_2 \neq 1 \end{cases} \tag{36}$$

using (29) we obtain

$$\text{sign}([w]_s + [v]_{x_2} + b) = \begin{cases} 1 & \text{if } x_2 = 1 \\ -1 & \text{if } x_2 \neq 1 \end{cases} \tag{37}$$

and unfolding the cases we can rewrite (37) as the system of equations

$$\begin{cases} \text{sign}([w]_s + [v]_1 + b) = 1 \\ \text{sign}([w]_s + [v]_j + b) = -1 \quad \forall j \neq 1 \end{cases} \tag{38}$$

which means that the following inequalities must hold

$$\begin{cases} [w]_s + [v]_1 + b > 0 \\ [w]_s + [v]_j + b < 0 \quad \forall j \neq 1 \end{cases} . \tag{39}$$

Doing the same steps for $x_1 = t$ we obtain that these inequalities must also hold

$$\begin{cases} [w]_t + [v]_1 + b < 0 \\ [w]_t + [v]_j + b > 0 \quad \forall j \neq 1 \end{cases} \tag{40}$$

[1]The case $m = 1$ can instead be trivially proved to be linearly separable.

Finally we notice the following implications involving $[v]_1$:

$$\begin{cases} (39) \implies [v]_1 > [v]_j \ \forall j \neq 1 \\ (40) \implies [v]_1 < [v]_j \ \forall j \neq 1 \end{cases} \tag{41}$$

and there we find the wanted contradiction. □

So we can now understand that if we have a dataset with categorical features and we one-hot encode them, we cannot hope to always have linear separability of the one-hot encoded points in $\mathbb{R}^n$, with any generic labeling.

References

[1] Haneen Alsuradi, Wanjoo Park, and Mohamad Eid. "Explainable Classification of EEG Data for an Active Touch Task Using Shapley Values". In: *Lecture Notes in Computer Science*. Springer International Publishing, 2020, pp. 406–416.

[2] Rik Sarkar Benedek Rozemberczki. "The Shapley Value of Classifiers in Ensemble Games". In: *In Woodstock '18: ACM Symposium on Neural Gaze Detection* (2018).

[3] Javier Castro, Daniel Gómez, and Juan Tejada. "Polynomial calculation of the Shapley value based on sampling". In: *Computers & Operations Research* 36.5 (2009), pp. 1726–1730.

[4] Dheeru Dua and Casey Graff. *UCI Machine Learning Repository - Mushrooms*. 2017. URL: http://archive.ics.uci.edu/ml/datasets/Mushroom.

[5] Ramón Flores, Elisenda Molina, and Juan Tejada. "Evaluating groups with the generalized Shapley value". In: *4OR* 17.2 (2018), pp. 141–172.

[6] Amirata Ghorbani and James Zou. "Data Shapley: Equitable Valuation of Data for Machine Learning". In: *In International Conference on Machine Learning* (2019), pp. 2242–2251.

[7] Francesco Lorenzi and Giacomo De Lazzari. *Mushley*. Version 1.0.0. Jan. 2022. URL: https://github.com/gdelazzari/Mushley.

[8] Scott M Lundberg and Su-In Lee. "A Unified Approach to Interpreting Model Predictions". In: *Advances in Neural Information Processing Systems 30*. Ed. by I. Guyon et al. Curran Associates, Inc., 2017, pp. 4765–4774.

[9] Hanchuan Peng, Fuhui Long, and C. Ding. "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27.8 (2005), pp. 1226–1238.

[10] Lloyd S. Shapley. "A value for n-person games". In: *The Shapley Value*. Cambridge University Press, 1988, pp. 31–40.

[11] Xin Sun et al. "Using cooperative game theory to optimize the feature selection problem". In: *Neurocomputing* 97 (2012), pp. 86–93.