Australian Government
Department of Defence
Science and Technology

TECHNICAL REPORT

A Power Series Expansion of Feature Importance

T. L. Keevers [1]

[1] Joint and Operations Analysis Division
Defence Science and Technology Group

DST-Group-TR-3743

Produced by

Joint and Operations Analysis Division
Defence Science and Technology Group

DST Headquarters
Department of Defence F2-2-03
PO Box 7931
Canberra BC ACT 2610

www.dst.defence.gov.au
Telephone: 1300 333 362

EXECUTIVE SUMMARY

Statistics-based machine learning and artificial intelligence can enhance the capabilities of complex and critical systems, but they can also increase new risks; statistical models may fail to generalize to novel data or situations, and cause the overall system to malfunction. A key issue for these models is the need to generalize to previously unobserved data or situations. Failure to do so can have severe reputational, financial, or safety implications. Cross-validation is the de facto standard for assessing a model's generality and performance. However, as we have argued in an earlier technical report (DST-Group-TR-3576), the limitations of cross-validation are often underappreciated. It doesn't guard against the possibility of algorithmic bias, drift in the sampling distribution, adversarial inputs, or a number of other issues.

A more fundamental understanding of statistical models can promote greater trust from the user and improve model robustness to novel data and situations. We have developed a power series formulation of feature importance that explicitly identifies individual and interaction-type contributions. The decomposition quantifies the impact of information and provides insight into whether features provide complementary, independent, or redundant information. Our method complements alternative approaches, such as clustering and subset selection, and provides a unique measure of feature importance.

Measures of feature importance should be able to accommodate different contexts and topics of interest. Missing features will affect models in a number of ways: It can change the model's structure, performance, memory footprint, or computation time. Our framework is able to handle these attributes by substituting an appropriate scalar metric into its calculation. Likewise, feature importance can refer to the expected feature importance prior to data collection, or the impact of observing a particular feature value actually had on the model output. Again, our framework can naturally handle both situations by changing the sampling distribution it uses to calculate the loss or fidelity. The flexibility of the framework allows for meaningful comparisons between users who may have different contexts or aims for the feature importance calculations.

Our final contribution is to show the power series can be mapped to the well-known Shapley values. These provide a method of fairly distributing output in a coalition game, and are the only formulation that has a number of intuitively desirable properties (for machine learning these properties are local accuracy, missingness and consistency, described elsewhere). Shapley values are ambiguous as to how features interact with each other. Equal Shapley values for two features could indicate that the features are completely dependent upon each other, like in the exclusive-OR problems, or may indicate the features have the same impact on the model, but are completely independent. Our method provides a fine-grained view of how features interact and is able to resolve these kinds of ambiguities, and can be used in situations that may be inappropriate for

Shapley values. Our approach also motivates efficient calculation schemes that reduce the number of computations required from exponential to polynomial. This allows feature importance calculations to be scaled to large numbers of features. Our power series formulation is versatile, theoretically-grounded, and motivates efficient calculation schemes. The power series formulation extends Defence's ability to interpret data, and will support the effective generation and maintenance of sophisticated statistical models.

CONTENTS

FIGURES

TABLES

NOTATION

c	Coalition output
D	Sampling distribution
E	Expectation
F	Fidelity function
f	Features
g	Imputation function
I	Indicator function
i	Index
j	Number of features within a grouping
L	Loss function
n	Number of features
p	Probability
q	Players
r	Subset of players
v	Instance variable importance
v	Local variable importance
V	Global variable importance
x	Data
x_c	The complement of a subset of data
x_s	Subset of data
y	Dependent variable
z	A scalar parameter
α	A coefficient
β	A coefficient
ϕ	An angle
δ	Model
δ_f	Dirac delta function
ϵ	Error term
τ	A scalar parameter
θ	Set of possible models
ρ	Correlation
π	Permutation
Ω	Set of features in a compound feature
ω	Subset of Ω
λ	A scalar parameter

This page is intentionally blank

This page is intentionally blank

1. INTRODUCTION

Critical systems can be enhanced through the integration of complex statistical models, like neural networks for image recognition. However, these statistical models can also introduce unforeseen risks. Many statistical models are blackboxes that lack transparency, and it can be hard to predict what factors they use to generate their outputs. Cross-validation is the standard method for measuring the performance of a supervised statistical learning model. However, as we have discussed before [1], cross-validation is inherently limited in its ability to check if a model truly generalizes. It cannot effectively address potential modelling pitfalls such as algorithmic bias, missing data, drift in the sampling distribution, extrapolation of models to areas of low or no sample density, data leakage, adversarial inputs, causal interpretations of the model, or poor quality data. Clearly additional techniques for model validation are essential for critical systems.

Model validation can be improved through the use of interpretable models and methods [2, 3]. Interpretability is context-specific, dependent upon the user, desired outcomes, and domain of application. Roughly speaking, interpretability refers to the ability to understand how and why statistical models produce their observed outputs. Insight into the model behaviour can be achieved through the use of 'intrinsically' interpretable models, exploratory analysis, or numerical metrics.

A persistent difficulty with model interpretation is that there are often multiple coherent, but inconsistent, explanations for what features are important, sometimes known as the Rashomon effect [4]. This problem is further compounded by the multitude of methods for creating interpretable models that can be found in the literature. Often they use different principles, may provide inconsistent information, and there is no guidance for choosing which method is appropriate.

A related difficultly is that the feature importance will depend on what information is of interest. For example, feature importance may relate to how valuable it would be to collect information or it could describe - retrospectively - how a feature's value impacted on the model's output. These decisions are often made implicitly, causing a proliferation of similar, but inconsistent, measures for feature importance. While they are valid when their (implicit) assumptions are satisfied, they can provide misleading information when used outside of their appropriate context.

We introduce a power series representation of feature importance that quantifies the impact of information. It separates out the marginal contribution of features into individual and multi-feature contributions. Our power series formulation can be mapped to Shapley values while providing a finer grained view of feature importance. This can uncover subtleties, such as whether features interact together, like in the exclusive-OR problem, or if they are truly independent. Our framework is flexible and can address many problem formulations with a single theoretically-

grounded approach. Additionally, it provides a way for dramatically decreasing the number of model evaluations required to calculate Shapley values under some circumstances.

We develop the power series formulation, map them to Shapley values, and then contrast it with other feature importance formulations. In Section 2, we examine the different ways in which feature importance can be formulated and show how they can be accommodated within our framework. Extensions and applications of this methodology are explored in Section 3. In Section 4, the formal relationship between our power series formulation and Shapley values is established, followed by examples in which the power series outperforms conventional Shapley values. A comparison of the power series formulation with other approaches is presented in Section 5, including comparison against other Shapley value-based methods. Finally, concluding remarks are presented in Section 6.

The primary contributions in this technical report are:

- We lay out explicit criteria for defining feature importance.
- We develop a power series representation of feature importance.
- We identify several useful properties of the power series, such as rules for generating compound feature importance and the invariance of the feature importance under monotonic transformations.
- We show the power series can be transformed into Shapley values.

2. DEFINING FEATURE IMPORTANCE

Feature importance will be defined by its context and purpose. A series of subtle decisions must be made by the end user when deciding what formulation is appropriate. We briefly outline each of the main considerations (derived heuristically) below, and we show how they can be treated coherently with a single framework that we develop through later sub-sections. Other approaches to defining interpretability can be found in the references [2, 5, 6] and in section 5.

- **Metric.** Defining feature importance requires us to define what aspects of a model are important. Access to information will affect the structure of a statistical model and its predictive performance, although potentially to different degrees. For example, when there are correlated features, it is often possible to find several linear regressions that achieve similar predictive performance, but with different feature weights. Conversely, it may be possible to find models with seemingly similar structures that have significantly different predictive performance. There may even be other implications of including or removing key features, like changes in the memory required to store a model or the computational time required to use it. These implications differ from accuracy and fidelity as they are implementation-dependent and cannot be discussed purely in terms of the underlying statistical model. In any case, the metric should link changes in feature availability to changes in the model attributes of interest. In a medical context, we may care about recall – the proportion of true cases identified – because of the high cost of a false negative.

- **Model-dependence.** Feature importance can be defined intrinsically as the correlation between independent and dependent features for a (fixed) sampling distribution, or extrinsically by the impact a feature has on a particular model. These two formulations can produce diverging conclusions if the model has not learnt the sampling distribution to a reasonable approximation. In the medical domain, we may have an exact sampling distribution for a numerical model, a good approximation for the sampling distribution when undertaking a large-scale randomized control study, but we will probably need to rely on crude approximations when analysing data from early stage trials. Across each of these transitions, our analysis becomes more sensitive to the details of our modelling.

- **Baseline for comparison.** Feature importance can measure the impact of missing features during training, testing, or both. Many models are unable to handle missing values, which means an appropriate baseline needs to be determined. Two natural methods for accommodating missing data are imputation schemes and the construction of alternative models that use a subset of features. But these methods can be under-defined, so the apparent feature importance can depend on arbitrary factors decided by the user. If performing a medical diagnosis, we might impute the model with the most common value.

Again, the baseline needs to be domain-relevant.

- **Prospective or retrospective?** Feature importance could refer to the expected importance prior to inspecting the feature's value, or the conditional importance after the feature was observed. Both of these might be relevant. For instance, we could consider an x-ray as a feature for medical diagnosis. There are health and monetary costs for performing an x-ray, and we may want to compare these costs against the prospect of improved diagnosis. After the x-ray is performed we may want to use the information retrospectively to justify the diagnosis of a bone fracture or sprain. It would be natural to treat an x-ray as an important feature for a suspected broken leg but not for a headache. Ideally our measure of feature importance should address both prospective and retrospective feature importance, while also accounting for the other information available.

We progressively address these considerations throughout the technical report.

2.1. Instance, Local, and Global Loss

Missing features will affect both the model's output and its predictive performance but potentially in different ways. The predictive performance is captured through a performance metric or loss L, which is commonly accuracy for classification tasks or root-mean square error for regression tasks. Alternatively, we can examine how missing data changes the output of a model through a fidelity function F. A fidelity function takes the outputs of two models as its inputs and measures their similarity. Unlike performance metrics, there are few conventions for what fidelity functions are appropriate. For classification tasks an obvious default would be a 0-1 function that produces 1 if and only if the outputs of two models match. Fidelity functions measure consistency, so there may be some cases where two models strongly agree with each other, but do not provide useful predictions.

We will soon define three types of feature importance, which relate the change in loss or fidelity to the presence or absence of features. We will look at losses for different scenarios (individual data points, locally, and globally) and then use them to generate feature importance. But to do this, we need to first define losses for each of these cases (fidelity can be treated by slightly modifying the equations below).

We use x to represent a complete set of features. Features within this set can include boolean values (a cough is present or not), ordinals (a subjective rating of pain), continuous values (a patient's temperature), or potentially more exotic features. We will generally treat our features as continuous values for the examples provided. This is primarily an aesthetic choice, and the integrals can be trivially replaced by summations when discrete features are present. We also use x_s to represent the subset of features that were observed, and x_c to represent the complement, the

unobserved features. x_s and x_c do not have to represent data that is missing from the data set or physically inaccessible; they are computational tools to counter-factually understand how information from specific feature values contribute to the model's output.

The loss for an individual instance depends on the model output, the dependent variable y, and the observed data x_s,

$$L(y, \delta(x_s)), \tag{1}$$

where δ is a statistical model. An example of this loss would be:

- L is the error rate in medical diagnosis (for example, does this patient have the flu?)
- y represents the true state (the patient is healthy)
- x_s are the observed features (a cough is present and the patient's temperature is 37°C)
- δ is the doctor's medical diagnosis (the patient has the flu)

Since the doctor has misdiagnosed the patient, there is a loss of 1 for this case.

While x_s and y will be correlated, there may still be some variation in y. For example, some patients with a cough and normal temperature will have the flu, and some won't. To capture the average model performance when a particular set of features is observed it is more appropriate to look at the expected local loss, which is given by

$$E_{Y|x_s}[L(y, \delta(x_s))] = \int L(y, \delta(x_s))\, p(y = Y|x_s) dY. \tag{2}$$

This provides the average misdiagnosis rate for a patient with cough and normal temperature in our example.

Similarly, the expected global loss describes the long-run model performance, and is given by

$$\begin{aligned} E_{X_S,Y}[L(y, \delta(x))] &= \int L(y, \delta(x_s))\, p(x_s = X_S, y = Y)\, dY\, dX_s \\ &= \int L(y, \delta(x_s))\, p(y = Y|x_s)\, p(x_s = X_s)\, dY\, dX_s, \end{aligned} \tag{3}$$

where the capital X, X_c, X_s and Y are used to represent random variables. In a medical context, the global expected loss would average over all combinations of patients (cough/no cough and normal temperature/abnormal temperature) to provide an overall misdiagnosis rate.

Sometimes it is easier to use a vector of indicator variables I to identify what features are available, rather than the values themselves,

$$x_i \in x_s \iff I_i = 1. \tag{4}$$

The vector of indicator variables will be used in some later exposition.

2.2. Baseline for Comparison

Equations 1, 2, and 3 express the change in the loss given different feature information. However, many models cannot natively handle missing values, seemingly leaving the expressions undefined. This requires the model (δ) to be somehow extended to incomplete feature sets.

Partial and complete data are connected by the relationship

$$p(y|x_s) = \int p(y|x_s, x_c)p(x_c = X_c|x_s)dX_c, \tag{5}$$

where the left-hand side represents the conditional probability with fewer features, and the right-hand side represents the standard conditional probability with imputed values. This suggests feature importance can be generalized by working with the partial data (the left-hand side) or attempting to impute the missing values (the right-hand side).

Imputation schemes fill in missing values with a 'best guess'. A probability density could theoretically be generated, but is rarely seen in practice [7]. In the simplest case imputation might be the mean value for a numerical feature or the mode for a categorical feature. The comparison can then be done between the model with the complete data and the model with imputed values. In many cases these imputations can be done quickly, which allows scaling to large data sets. However, the imputation scheme must be chosen by the user and there is no obvious mechanism for generating or selecting a particular imputation scheme among several possibilities. This means the comparison model is non-unique, and arbitrary decisions can impact on the (apparent) feature importance.

Alternatively, it is possible to work with the subset data x_s directly and generate a new model.

As with the imputation scheme, there is a degree of arbitrariness in how to select parameters for the data-restricted models. For example, for neural networks there may be a desire to adjust the learning rate, strength of the regularization, and so on in response to the number of observed features. The calculated feature importance (as opposed to the ideal Bayes-optimal feature importance) will be sensitive to these adjustments, even though the parameters are not related to the underlying sampling distribution. Training a series of new models can also be impractical for large data sets, requiring an alternative approach.

These two approaches will be identical for the *Bayes-optimal* model because the conditional probability $p(x_c|x_s)$ is either explicitly (imputation) or implicitly (retraining) summed over. The Bayes-optimal model provides the best possible predictive performance,

$$\delta_{Bayes} = \underset{\delta\in\theta}{\operatorname{argmin}} E_{x,y\sim D}[L(y,\delta(x))], \tag{6}$$

where D is the sampling distribution and θ is the set of possible models and we have assumed L is a loss function (so small values are better). This definition can easily accommodate missing data by changing x to encompass a different set of features. Likewise, a *Bayes-faithful* model can be defined by

$$\delta_{BF} = \underset{\delta'\in\theta}{\operatorname{argmax}} E_{x,y\sim D}[F(\delta(x_s,x_c),\delta'(x_s))] \tag{7}$$

$$= \underset{\delta'\in\theta}{\operatorname{argmax}} \int F(\delta(x_s,x_c),\delta'(x_s))p(x_c = X_c|x_s)p(x_s = X_s)\,dX_c\,dX_s. \tag{8}$$

where we assume F is maximized when $\delta = \delta'$. The Bayes models have no free parameters and the feature importance will be non-negative for every feature.

The Bayes-optimal model provides an objective measure of feature importance. It is derived solely from the sampling distribution and is therefore independent of any arbitrary user choices. By construction it also provides an upper bound for model performance. However, Bayes-optimal models are rarely accessible in practice, and their behaviour can diverge from that seen with realistic models. First, removing a feature from a model can improve performance [8], leading to a nominal negative feature importance for predictive performance. We do not know of any cases in which replacing real data values with imputed values has improved the model performance, although it could conceivably occur if the model has overfitted to the feature in question. Second, data imputation and retraining the model will rarely provide the same results. Data imputation assumes that the missing values were observed for the training data, while retraining a model

can be performed regardless of whether data is missing during the test or training phases. One could conceivably think of imputation as measuring the feature importance for a fixed statistical model and retraining as measuring feature importance for a learning algorithm, although we are interpretation-agnostic.

2.3. Representing Feature Importance as a Power Series

The expected loss of a model will usually depend non-linearly on the observed features. One way to explicitly represent this non-linearity is through a series expansion. To accomplish this, we propose expressing the joint dependence between features by a power series. For this global feature importance this is

$$E_{X,Y}[L(y,\delta(x))] = V_{\emptyset} + \sum_{i=0}^{n-1} V_i \cdot I_i + \sum_{i=1}^{n-1}\sum_{j=0}^{i-1} V_{ij} \cdot I_i \cdot I_j + \sum_{i=2}^{n-1}\sum_{j=1}^{i-1}\sum_{k=0}^{j-1} V_{ijk} \cdot I_i \cdot I_j \cdot I_k \ldots \quad (9)$$

The first term is the expected loss given no features are available, it is the base rate. The other V terms represent coefficients in the power series, and I (again) acts as an indicator variable. The higher-order terms account for interactions. We will shortly demonstrate how to calculate these terms. For example, the second-order term V_{ij} is given by

$$V_{ij} = E_{X,Y}[L(y,\delta(x)|I_i, I_j = 1)] - V_i - V_j - V_{\emptyset}, \quad (10)$$

Other definitions of feature importance also capture the general concept of joint importance [9], although their precise formulations have some variation. Our decomposition defines the interaction terms relative to the baseline of no features, while other popular formulations define the interaction relative to the possible coalitions that can be formed [10–12]. One benefit of our approach is that the interaction coefficients are unaffected by the addition or removal of features that do not directly take part in the interaction.

2.4. Defining Feature Importance

In equation 9, we defined feature importance through a power series. For just two features, we can define the interaction using equation 10. We can easily generalize this to cover instance (v), local (v), or global (V) feature importance. We define the terms in the power series using a recurs-

ive relationship for each case:

$$v_S(y,x) = L(y,\delta(x_S)) - \sum_{s\subsetneq S} v_s(y,x), \tag{11}$$

$$v_S(x) = E_{Y|x_s}[L(y,\delta(x_s))] - \sum_{s\subsetneq S} v_s(x), \tag{12}$$

and

$$V_S = E_{Y|I_{s\in S}}[L(y,\delta(X_s))] - \sum_{s\subsetneq S} V_s, \tag{13}$$

and the empty set terms in each are:

$$v_\emptyset(y,x) = L(y,\delta(\emptyset)), \tag{14}$$

$$v_\emptyset(x) = E_{Y|X}[L(y,\delta(\emptyset))], \tag{15}$$

and

$$V_\emptyset = E_Y[L(y,\delta(\emptyset))]. \tag{16}$$

The recursive relationship arises from defining interactions as the marginal contribution after removing all lower-order effects. The power series terms have the same structure as the Harsanyi dividend in cooperative game theory [13].

For the purposes of computing the feature importance, it can be assumed that L and δ are known, as well as x and y, if applicable. This allows the terms to be formally calculated, although the process of generating δ and evaluating $\delta(x)$ can be quite computationally expensive in practice. The base case for the induction, for example $v_\emptyset(y,x)$, can always be immediately computed, providing the base step for the computation. We can expand out the terms for a model with three features (x_0, x_1, x_2):

The baseline term is simply

$$\nu_{\emptyset}(y,x) = L(y,\delta(x_S = \{\emptyset\})). \tag{17}$$

The first-order terms are

$$\nu_{x_0}(y,x) = L(y,\delta(x_S = \{x_0\})) - \underbrace{L(y,\delta(x_S = \{\emptyset\}))}_{\text{baseline term}}, \tag{18}$$

$$\nu_{x_1}(y,x) = L(y,\delta(x_S = \{x_1\})) - \underbrace{L(y,\delta(x_S = \{\emptyset\}))}_{\text{baseline term}}, \tag{19}$$

and

$$\nu_{x_2}(y,x) = L(y,\delta(x_S = \{x_2\})) - \underbrace{L(y,\delta(x_S = \{\emptyset\}))}_{\text{baseline term}}. \tag{20}$$

The second-order interaction terms are

$$\nu_{x_0,x_1}(y,x) = L(y,\delta(x_S = \{x_0,x_1\})) - \underbrace{(\nu_{x_0}(y,x) + \nu_{x_1}(y,x) + \nu_{x_2}(y,x))}_{\text{first-order terms}} - \underbrace{\nu_{\emptyset}(y,x)}_{\text{baseline term}}, \tag{21}$$

$$\nu_{x_0,x_2}(y,x) = L(y,\delta(x_S = \{x_0,x_2\})) - \underbrace{(\nu_{x_0}(y,x) + \nu_{x_1}(y,x) + \nu_{x_2}(y,x))}_{\text{first-order terms}} - \underbrace{\nu_{\emptyset}(y,x)}_{\text{baseline term}}, \tag{22}$$

and

$$\nu_{x_1,x_2}(y,x) = L(y,\delta(x_S = \{x_1,x_2\})) - \underbrace{(\nu_{x_0}(y,x) + \nu_{x_1}(y,x) + \nu_{x_2}(y,x))}_{\text{first-order terms}} - \underbrace{\nu_{\emptyset}(y,x)}_{\text{baseline term}}. \tag{23}$$

Finally, the third-order interaction term is

$$\nu_{x_0,x_1,x_2}(y,x) = L(y,\delta(x_S=\{x_0,x_1,x_2\})) - \underbrace{(\nu_{x_0,x_1}(y,x)+\nu_{x_0,x_2}(y,x)+\nu_{x_1,x_2}(y,x))}_{\text{second-order terms}} - \underbrace{(\nu_{x_0}(y,x)+\nu_{x_1}(y,x)+\nu_{x_2}(y,x))}_{\text{first-order terms}} - \underbrace{\nu_{\emptyset}(y,x)}_{\text{baseline term}}. \quad (24)$$

The process can be extended to an arbitrary number of features. There will be a total of 2^n power series coefficients, where n is the number of features.

2.5. Relationship Between Forms of Feature Importance

The three forms of feature importance are related to each other through their expected values,

$$\nu_S(x) = E_Y[\nu_S(y,x_S)] = \int \nu_S(y,x_S)p(y=Y)dY, \quad (25)$$

and

$$V_S = E_X[\nu_S(x)] = \int \nu_S(x)p(x=X)dX. \quad (26)$$

We can also think of these forms as being *temporally coupled*: the global feature importance is how important we expect a feature to be (say, the presence or absence of a cough) before we know its value; the local feature importance is how important we expect a feature to be after we know its value (the patient has a cough) but before we know the label/predicted quantity y (the patient is healthy); the instance feature importance is how important we expect a feature to be after we know its value and the label/predicted quantity y (the cough was a red herring in this case, we incorrectly used it to diagnose flu).

While there are three forms of feature importance, we can usually decide which form is relevant based on the information at hand and the goals we want to accomplish. If we want to justify why a model made a certain prediction, we would look at local feature importance - "the neural network predicted an elephant because of the trunk". In contrast, we may want to develop a short health screening tool. This might consist of a small number of questions: "do you have a fever? have you recently travelled overseas?" We could justify choosing a subset of possible questions using the global feature importance.

Table 1 The probability of observing a data point (x, y). There is a single feature x = 0 or 1 and three classes (y = 0, 1, or 2). The marginal probabilities are given along the final row and column.

	x=0	x=1	$p(y)$
y=0	0.30	0.00	0.30
y=1	0.09	0.24	0.33
y=2	0.21	0.16	0.37
$p(x)$	0.60	0.40	

2.6. A Three Class Problem

When some features are missing, it is possible to train a new model that is as faithful as possible to the original model, or to instead focus on minimizing the loss for the simpler model. Perhaps surprisingly, these two objectives can be in tension even when the original model is optimal. To illustrate this tension, we consider a simple classification problem involving one feature (x = 0 or 1) and three classes (y = 0, 1 or 2), with the probabilities provided in Table 1. The performance metric is accuracy.

From the table we see that the Bayes classifier (δ_{Bayes}) will predict classes using the rule

$$\delta_{Bayes}(x) = \begin{cases} 0 & \text{if } x = 0 \\ 1 & \text{if } x = 1 \\ 2 & \text{if } x \text{ is missing} \end{cases} \tag{27}$$

The classifier will predict the class '0' 60% of the time and class '1' 40% of the time, with an overall accuracy of 54% ($p(y = 0|x = 0)p(x = 0) + p(y = 1|x = 1)p(x = 1) = 0.30 + 0.24 = 0.54$). If we remove information about x we can either train a new classifier to be as faithful to the original classifier as possible (δ_0), or to maximize its accuracy (δ_1). The most faithful classifier will always predict '0', which has a fidelity of 0.6 and an accuracy of only 30%. In contrast, the Bayes-optimal classifier when no feature information is available will always predict the class '2' with a fidelity of 0 and an accuracy of 37% ($p(y = 2|x = 0)p(x = 0) + p(y = 2|x = 1)p(x = 1) = 0.21 + 0.16 = 0.37$). In this situation we find a trade-off between fidelity and accuracy. Which of these quantities we want to maximize will depend on the question of interest. If we want to understand *how* the model makes its decision, the fidelity is probably of interest, while the accuracy will probably be more useful if we are trying to identify what data we want to collect.

2.7. Example of Linear Regression

We demonstrate the power series decomposition for a simple linear regression with two potentially correlated features (x_0, x_1) drawn from a bivariate Gaussian distribution. The true relation-

ship is

$$y = \alpha x_0 + \beta x_1 + \epsilon, \tag{28}$$

where ϵ is a Gaussian error term with mean zero and a standard deviation of σ, and

$$p(x_0, x_1) = \frac{1}{2\pi\sigma_0\sigma_1\sqrt{1-\rho^2}} \exp\left(-\frac{1}{2(1-\rho^2)}\left(\frac{x_0^2}{\sigma_0^2} + \frac{x_1^2}{\sigma_1^2} - \frac{2\rho x_0 x_1}{\sigma_0\sigma_1}\right)\right), \tag{29}$$

where the terms have their standard definitions.

When a square-error loss function is used, the optimal prediction is given by

$$\delta(x) = \begin{cases} 0 & \text{if } x_s = \emptyset, \\ \alpha x_0 + \beta\rho\frac{\sigma_1}{\sigma_0}x_0 & \text{if } x_s = \{x_0\}, \\ \alpha\rho\frac{\sigma_0}{\sigma_1}x_1 + \beta x_1 & \text{if } x_s = \{x_1\}, \\ \alpha x_0 + \beta x_1 & \text{if } x_s = \{x_0, x_1\}, \end{cases} \tag{30}$$

and can be derived by noting that the conditional expectation of x_0 given x_1 is

$$E(X_0|X_1 = x_1) = \rho\frac{\sigma_0}{\sigma_1}x_1 \tag{31}$$

and vice versa for x_1 given x_0.

The error for a specific instance is given trivially by

$$(y - \delta(x))^2 \tag{32}$$

which can be broken down for each case:

$$L(y,\delta(x)) = \begin{cases} (\alpha x_0 + \beta x_1 + \epsilon)^2 & \text{if } x_s = \emptyset, \\ (\beta(x_1 - \rho\frac{\sigma_1}{\sigma_0}x_0) + \epsilon)^2 & \text{if } x_s = \{x_0\}, \\ (\alpha(x_0 - \rho\frac{\sigma_0}{\sigma_1}x_1) + \epsilon)^2 & \text{if } x_s = \{x_1\}, \\ \epsilon^2 & \text{if } x_s = \{x_0, x_1\}, \end{cases} \tag{33}$$

where the conditionals on the right-hand side indicate what feature information is available. The systematic and random error terms can sometimes cancel each other out, which means the model performs worse for some specific instances when more information is available.

The expected error of a model for a fixed data point $x = (x_0, x_1)$ can be calculated by taking the expectation of the terms above. The expected error is given by the sum of a systematic bias-squared term (the first term in the equation below) and the variance from the random and uncorrelated error (the second term) [8]. While the systematic and random error terms can cancel on occasions, we see that on average they will add together,

$$E_Y[L(y,\delta(x_s))] = \begin{cases} (\alpha x_0 + \beta x_1)^2 + \sigma^2 & \text{if } x_s = \emptyset, \\ \beta^2(x_1 - \rho\frac{\sigma_1}{\sigma_0}x_0)^2 + \sigma^2 & \text{if } x_s = \{x_0\}, \\ \alpha^2(x_0 - \rho\frac{\sigma_0}{\sigma_1}x_1)^2 + \sigma^2 & \text{if } x_s = \{x_1\}, \\ \sigma^2 & \text{if } x_s = \{x_0, x_1\}, \end{cases} \tag{34}$$

The local feature importance can be calculated from combinations of the above terms,

$$v(x_0,x_1) = \begin{cases} v_\emptyset(x_0,x_1) = (\alpha x_0 + \beta x_1)^2 + \sigma^2, \\ v_0(x_0,x_1) = \beta^2(x_1 - \rho\frac{\sigma_1}{\sigma_0}x_0)^2 - (\alpha x_0 + \beta x_1)^2, \\ v_1(x_0,x_1) = \alpha^2(x_0 - \rho\frac{\sigma_0}{\sigma_1}x_1)^2 - (\alpha x_0 + \beta x_1)^2, \\ v_{01}(x_0,x_1) = (\alpha x_0 + \beta x_1)^2 - \beta^2(x_1 - \rho\frac{\sigma_1}{\sigma_0}x_0)^2 - \alpha^2(x_0 - \rho\frac{\sigma_0}{\sigma_1}x_1)^2. \end{cases} \tag{35}$$

$\delta(x)$ tries to use the correlation between x_0 and x_1 to reduce the error, which means there is a coupling between the two features in the local importance, even when only the first-order terms are considered.

It is instructive to calculate the limiting case $\alpha \to 0$,

$$\lim_{\alpha\to 0} v(x_0, x_1) = \begin{cases} v_\emptyset(x_0, x_1) = \beta^2 x_1^2 + \sigma^2, \\ v_0(x_0, x_1) = \beta^2 \rho^2 \frac{\sigma_1^2}{\sigma_0^2} x_0^2 - 2\rho \frac{\sigma_1}{\sigma_0} \beta^2 x_0 x_1, \\ v_1(x_0, x_1) = -\beta^2 x_1^2, \\ v_{01}(x_0, x_1) = -\beta^2 \rho^2 \frac{\sigma_1^2}{\sigma_0^2} x_0^2 + 2\rho \frac{\sigma_1}{\sigma_0} \beta^2 x_0 x_1. \end{cases} \tag{36}$$

Even though the value of x_0 does not impact the loss function directly (because α = 0), it still provides some value through its correlation with x_1. This is seen by the non-zero expressions for v_0 and v_{01}.

Finally, the expected global error is calculated by integrating over the distribution of possible values and weighting them by an appropriate probability density. It is given by

$$E_{X,Y}[L(y, \delta(x))] = \begin{cases} \alpha^2\sigma_0^2 + 2\alpha\beta\rho\sigma_0\sigma_1 + \beta^2\sigma_1^2 + \sigma^2 & \text{if } X_s = \{\emptyset\}, \\ (1-\rho^2)\beta^2\sigma_1^2 + \sigma^2 & \text{if } X_s = \{X_0\}, \\ (1-\rho^2)\alpha^2\sigma_0^2 + \sigma^2 & \text{if } X_s = \{X_1\}, \\ \sigma^2 & \text{if } X_s = \{X_0, X_1\}, \end{cases} \tag{37}$$

where X_s on the right-hand side again indicates which features are available.

As we are using a loss function, a negative feature importance indicates a reduction in the global expected loss, the convention would be the opposite for an objective function that we are trying to maximize,

$$V = \begin{cases} V_\emptyset = \alpha^2\sigma_0^2 + 2\alpha\beta\rho\sigma_0\sigma_1 + \beta^2\sigma_1^2 + \sigma^2, \\ V_0 = -(\alpha^2\sigma_0^2 + 2\alpha\beta\rho\sigma_0\sigma_1 + \rho^2\beta^2\sigma_1^2), \\ V_1 = -(\rho^2\alpha^2\sigma_0^2 + 2\alpha\beta\rho\sigma_0\sigma_1 + \beta^2\sigma_1^2), \\ V_{01} = 2\alpha\beta\rho\sigma_0\sigma_1 + \rho^2(\alpha^2\sigma_0^2 + \beta^2\sigma_1^2). \end{cases} \tag{38}$$

The redundant information is related to the variation (σ_0, σ_1) directly explained by each feature and the correlation (ρ) between them.

2.8. When is Feature Importance not Defined?

Our framework provides a broadly applicable method for determining how features contribute to a model's output. We briefly look at some situations in which our framework may breakdown or is not applicable.

Diverging loss. Our approach assumes the expected loss is bounded. This assumption can pathologically fail for heavy-tailed distributions. A classical example where the loss diverges is the mean-square error for a Cauchy distribution, which doesn't even have a properly defined mean.

Interventions. Sometimes we may want to calculate the value of an intervention, such as the change in patient life expectancy given treatment for a disease. We could then consider factors like age or gender to identify interaction terms. The treatment feature is externally controlled and is not well-described by our statistical approach, so cannot be immediately addressed in our framework.

Continuous feature inclusion. We treated features as strictly included or excluded in our framework. In some situations, it might be more natural to consider partial inclusion. Consider the situation in which we are trying to estimate $y \sim N(0, z^{-1})$ under a mean-square error and we have a single feature $x \sim N(y, \tau^{-1})$, where τ is a measurement precision we can control. Then after sampling x, the probability distribution for y is

$$p(y|x) = N\left(\frac{\tau}{z+\tau}x, \frac{1}{z+\tau}\right), \tag{39}$$

and the mean-square error drops from

$$z^{-2} \tag{40}$$

to

$$(z+\tau)^{-2}. \tag{41}$$

We could treat the feature importance by using the two equations above, but this ignores our ability to choose τ. Unlike the interventions above, where we are directly changing the properties of the system, we are now only considering the value of a non-interactive, passive measurement.

Unlimited features. Some kinds of non-parametric models can possess arbitrarily large numbers of features. A basic example is a nearest-neighbour model. While we can always calculate the feature importance for a finite data set, this may not be defined in the asymptotic limit.

Undefined feature censoring. We have treated features as either present or absent. It is not clear that this distinction is useful for all types of data sets. For example, if we have a sample of text - "the car is red" - we may identify the colour of an object from the word 'red'. Should the absent case be represented by replacing red with a star '*', by its removal from the sentence, or by some other kind of censoring?

3. EXTENSIONS AND APPLICATIONS

The power series framework can be applied to a variety of problems. In this section, we develop a deeper understanding of the power series. We first look at how feature transformations affect the power series (subsections 3.1, 3.2, 3.3), then we examine the potential for visualization (subsection 3.4), some summary statistics for feature importance are given (subsection 3.5), provide some details for how we may want to calcualte the feature importance in practice for Gaussian mixture models (subsection 3.6), and then we conclude by looking at how the power series can be related to influential data points (subsection 3.7) and identifying what features are sensitive to censoring in the context of adversarial attacks (subsection 3.8).

3.1. Monotonic Transformations

The local and global feature importance is unaffected by any monotonic transform that operates on a single feature. To show this we apply an arbitrary monotonic transform and show that the expressions for the loss for a single data point, the local expected loss, and the global expected loss (equations 1, 2, and 3) are invariant. Since the feature importance is derived from these equations, it follows that the feature importance is also invariant.

To begin, we select an arbitrary feature x_i and transform it using a monotonic transform t,

$$w_i = t(x_i). \tag{42}$$

We assume the probability density function and transforms are smooth and 'well-behaved' so they can be integrated and differentiated. The loss for a single data point x is

$$L(y, \delta(x)). \tag{43}$$

The loss can be re-expressed as

$$L(y, \delta(x_{-i}, x_i)) = L(y, \delta(x_{-i}, t^{-1}(w_i)), \tag{44}$$

where we have split x into the single feature x_i and the remaining features x_{-i}. The original

model cannot operate on w_i so we define a new model δ' that satisfies

$$L(y, \delta(x_{-i}, t^{-1}(w_i)) = L(y, \delta'(x_{-i}, w_i)). \quad (45)$$

We can always find such a δ'; one way would be to first apply the transform t to w_i to recover x_i and then use the original model δ. This equality means the loss for any data point (equation 1) is unaffected by a monotonic transformation of one feature, and is true by definition for the Bayes-optimal model or by construction in general.

The expected local loss (equation 2) is calculated by integrating over the conditional probability,

$$\int L(y, \delta(x_s))\, p(y = Y|x_s) dY = \int L(y, \delta(x_{-i}, t^{-1}(w_i)))\, p_{Y|X_{-i},X_i}(y = Y|x_{-i}, t^{-1}(w_i)) dY \quad (46)$$

$$= \int L(y, \delta'(x_{-i}, w_i))\, p_{Y|X_{-i},W_i}(y = Y|x_{-i}, w_i) dY. \quad (47)$$

The conditional probabilities implicitly account for which feature (x_i or w_i) is used. Information is preserved because we apply a monotonic transformation,

$$p(y|w_i) = \int p_{Y|X_i}(y|x_i) p_{X_i|W_i}(x_i = X_i|w_i)\, dX_i \quad (48)$$

$$= \int p_{Y|X_i}(y|x_i)\, \delta_f(t^{-1}(w_i = W_i))\, dW_i = p_{y|x_i}(y|x_i), \quad (49)$$

where δ_f is the Dirac delta function (not the statistical model). This demonstrates that the local feature importance is invariant.

Finally, the global expected loss (equation 3) is

$$E_{X,Y}[L(y, \delta(x))] = \int L(y, \delta(x))\, p_{Y|X}(y = Y|x)\, p_x(x = X)\, dX\, dY \quad (50)$$

$$= \int L(y, \delta(x_{-i}, x_i))\, p_{Y|X_{-i},X_i}(y = Y|x_{-i}, x_i)\, p_{X_{-i}|X_i}(x_{-i} = X_{-i}|x_i)\, p(x_i = X_i)\, dX_{-i}\, dX_i\, dY \quad (51)$$

$$\begin{aligned} &= \int L(y, \delta(x_{-i}, t^{-1}(w_i)))\, p_{Y|X_{-i},X_i}(y = Y|x_{-i}, t^{-1}(w_i)) \\ &p_{X_{-i}|X_i}(x_{-i} = X_i|t^{-1}(w_i))\, p(t^{-1}(w_i) = t^{-1}(W_i))\, \frac{dx_i}{dw_i}\, dX_{-i}\, dW_i\, dY. \end{aligned} \quad (52)$$

$$= \int L(y, \delta'(x_{-i}, w_i))\, p_{Y|X_{-i},W_i}(y = Y|x_{-i}, w_i)\, p_{X_i|W_i}(x_{-i} = X_{-i}|w_i)\, p(w_i = W_i)\, dX_{-i}\, dW_i\, dY, \quad (53)$$

$$= E_{\delta'|I'}[L], \quad (54)$$

where I' is the inclusion of modified features, and we used the $\frac{dx_i}{dw_i}$ term to transform the probability term of the lone feature from $p(x_i)$ to $p(w_i)$. The global expected loss is the same, whether it is defined in terms of x_i or w_i, so it is also invariant.

This demonstrates that the feature importance is invariant under monotonic transformations. A similar proof applies to the discrete case for bijective mappings for a single feature. The invariance property, while not immediately obvious, informally means that the feature importance is independent of the units used for expressing a feature. There are other formulations of feature importance for which invariance does *not* apply. Virtually any method that uses gradients to determine local feature importance (for example [14]) is not invariant, even when a monotonic transform is used.

3.2. Feature Transformation

The feature importance can change dramatically after applying a multi-feature transformation. To demonstrate this, we examine the change in feature importance when the two correlated components of a bi-normal distribution are replaced with two uncorrelated components. This can be achieved by applying an appropriate rotation with angle ϕ to the features from equation 29. The

two new features u_0 and u_1 are related to the original features by

$$u_0 = x_0 \cos\phi + x_1 \sin\phi, \tag{55}$$

and

$$u_1 = -x_0 \sin\phi + x_1 \cos\phi. \tag{56}$$

The standard deviations of these features are

$$\sigma_{u0} = \frac{1}{2}[\sigma_0 + \sigma_1 + \sqrt{(\sigma_0 + \sigma_1)^2 - 4\sigma_0\sigma_1(1-\rho^2)}] \tag{57}$$

and

$$\sigma_{u1} = \frac{1}{2}[\sigma_0 + \sigma_1 - \sqrt{(\sigma_0 + \sigma_1)^2 - 4\sigma_0\sigma_1(1-\rho^2)}], \tag{58}$$

and now the two features are statistically independent. The feature importance can be calculated as before. For brevity, we only provide the global feature importance,

$$V = \begin{cases} V_\emptyset = (\alpha\cos\phi + \beta\sin\theta)^2\sigma_{u0}^2 + (-\alpha\sin\phi + \beta\cos\theta)^2\sigma_{u1}^2, \\ V_0 = -(\alpha\cos\phi + \beta\sin\phi)^2\sigma_{u0}^2, \\ V_1 = -(-\alpha\sin\phi + \beta\cos\phi)^2\sigma_{u1}^2, \\ V_{01} = 0. \end{cases} \tag{59}$$

It can be seen that the interaction term V_{01} is now completely suppressed, as expected, and this makes the interpretation of the features much simpler. The local feature importance in the transformed case bears little resemblance to the values in the untransformed case. Feature importance is related to the effect of removing a feature. Removing u_0 or u_1 reports partial information about both x_0 and x_1 because of the interaction between the features, and this means the effect on the expected loss is non-linear.

3.3. Combined Features

The power series decomposition can be applied to data with modified features, and thereby increase the types of problems that can be addressed. For example, humans wouldn't explain images in terms of individual pixels; they would identify a face from the presence of abstract objects like an eye or a nose. It is possible to represent more abstract or complex entities within our framework. For example, replacing a single pixel as a feature by a cluster of features.

There is a simple relationship between the feature importance for 'compound' features and those of their constituent features from which they're constructed. Consider two features, i and j, that have been grouped into a compound feature, $\{i,j\}$, then we can immediately deduce that the first-order feature importance of the compound feature is given by the first-order contributions of features i and j, as well as their interaction term. Mathematically this can be expressed by

$$V_{\{i,j\}} = V_i + V_j + V_{ij}. \tag{60}$$

Similarly, elementary reasoning shows that the second-order interaction with a feature k is given by

$$V_{\{i,j\}k} = V_{ik} + V_{jk} + V_{ijk}. \tag{61}$$

The higher-order terms and interactions between compound features follow a similar pattern. If the set of features Ω forms a compound feature, then the importance of this compound feature is related to the importance of constituent features by expanding out the interactions with each possible subset ω,

$$V_{\{\Omega\}j} = \sum_{\omega \subsetneq \Omega, \omega \neq \emptyset} V_{\omega j}. \tag{62}$$

Equation 62 provides an elegant method for aggregating features together and avoids elaborate recomputation that is required with some other methods, notably Shapley values that we describe shortly.

3.4. Visualizing Feature Importance

Feature importance can be made more insightful and interpretable through visualization. There are a number of avenues for visualizing numerical quantities, like those given in equation 35 that

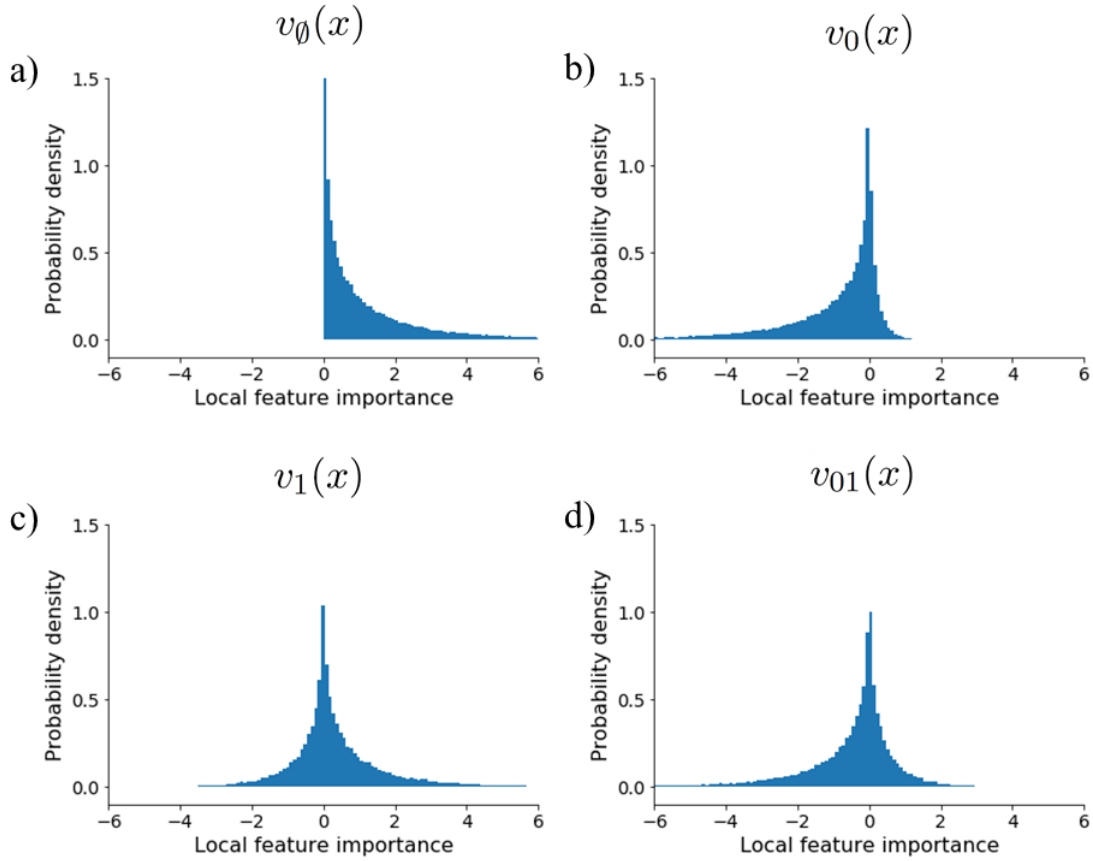


Figure 1 The local feature importance for $v_\emptyset$, v_0, v_1, and v_{01} for a linear model with points sampled from a bi-normal distribution and a mean-square error function. Positive (negative) indicates the feature increases (decreases) the model error. See text for more details.

describe the local feature importance for a two-dimensional Gaussian distribution. The local feature importance can be plotted as a marginal or conditional distribution by fixing the unknown features at specific values or averaging over the latent distribution. As a proof-of-concept we plot a histogram of local feature importance values in Figure 1. Recall, the general form of feature importance is given by equation 12, with a closed-form solution for the bi-Gaussian case in equation 35, and equation 29 provides the probability distribution for the features. Values for $v_\emptyset$, v_0, v_1, and v_{01} are plotted in 1 a-d, respectively.

To generate the histogram we assumed there was no random noise ($\sigma = 0$), that the components had equal variance ($\sigma_0 = \sigma_1 = 1$) and modest correlation ($\rho = 0.3$). The first feature had a stronger impact on the output than the second ($\alpha = 0.9, \beta = 0.44$). These properties are reflected in Figure 1. $v_\emptyset$ is always positive which is indicative of the baseline error, and has significant skew because Gaussian distributions have light tails. v_0 and v_1 are, on average, slightly negative, although there is significant spread. v_0 has a more negative value on average because $\alpha > \beta$. The model tries to use the correlation between x_0 and x_1 to reduce the average error. However,

in some cases the unknown feature has the opposite of the expected sign (the positive feature values), causing the performance to degrade. When this occurs, the interaction term (v_{01}) will be negative, indicating a synergistic effect. The positive feature values in this case are generated by the inherent stochasticity of the system, and are not the result of overfitting. Even for this simple example, basic visualization techniques like histograms can be used to insight into the interactions between features. In more complex cases, visualization may potentially assist in making blackbox models more interpretable.

3.5. Summary Statistics

In some situations we may want to provide some insight into the feature importance terms without resorting to a table of values. Visualization is one alternative method we have explored. Another avenue is summary statistics. In standard statistics, we are familiar with quantities like the mean, mode, and standard deviation that summarize the general attributes of statistical samples or distributions. These do not exist for interpretable machine learning, since the field is still emerging. We propose four summary statistics to fill this gap: the sufficient contribution, the necessary contribution, the minimum marginal contribution, and the maximum marginal contribution. These are defined in relation to the change in the expected global loss.

Sufficient. The sufficient contribution of feature X_i is λ if

$$E[L(y, \delta(\emptyset))] - E[L(y, \delta(X_s \cup X_i))] \geq \lambda \; \forall \; X_s \in X \setminus X_i. \tag{63}$$

This means that including feature X_i is our model is sufficient to decrease the expected loss by at least λ over the base rate.

Necessary. The necessary contribution of feature X_i is λ if

$$E[L(y, \delta(X \setminus X_i))] - E[L(y, \delta(X))] = \lambda. \tag{64}$$

This means that it is necessary to include feature X_i in our model, or the loss will be at least λ greater than for the Bayes-optimal model.

Minimum. The minimum marginal contribution of feature X_i is λ if adding X_i always decreases the

expected loss by at least λ,

$$E[L(y, \delta(X_s))] - E[L(y, \delta(X_s \cup X_i))] \geq \lambda \; \forall \; X_s \subset X \setminus X_i. \tag{65}$$

Maximum. The maximum marginal contribution of feature X_i is λ if adding X_i can decrease the expected loss by at most λ,

$$\sup_{X_s \in X \setminus X_i} \left(E[L(y, \delta(X_s))] - E[L(y, \delta(X_s \cup X_i))] \right) = \lambda. \tag{66}$$

These summary statistics can be adapted for local feature importance in a number of ways: we could apply the above definitions for a specific value of x; we could require the definitions to hold for all x in the sample space of X; or we might require a probabilistic guarantee. For example, that the reduction in loss is λ at least 95% of the time.

3.6. Calculating Feature Importance for a Gaussian Mixture

To calculate the local and global feature importance, we need to access the expected values. We may not be able to calculate these when we only have a finite number of samples, so some adjustments in our approach are required. The adjustments will depend on the type of model we are using. In this section, we briefly describe how the feature importance can be estimated for a Gaussian mixture model, although the principles we use are widely applicable to generative models.

We will assume we are using a cross-entropy loss function, and through some fitting procedure (maybe maximum likelihood), we have determined the vector of means for each class is a known $\hat{\mu}_i$ and the covariance matrix is a $\hat{\Sigma}_i$, where $i \in 0, \ldots, n-1$, and that the proportion of each class is $\hat{y}_i$.

We can calculate the conditional probability of y given x, $p_{\hat{\mu}_i, \hat{\Sigma}_i}(y_i|x)$, using readily available statistical packages.

The local expected loss for point x is

$$E_{Y|X=x}[L(y, \delta(x))] = \sum_{i=0}^{n-1} p_{\hat{\mu}_i, \hat{\Sigma}_i}(y_i|x) \log(p_{\hat{\mu}_i, \hat{\Sigma}_i}(y_i|x)). \tag{67}$$

The expected class for subsets of features can be calculated using the marginal distributions of the Gaussian components.

The expected loss using our generative model can then be fed into equation 12 to calculate the local feature importance. Likewise, we can calculate the instance and global feature importance. Since we are using the sampling distribution from our model, the values will differ from their true values. The deviation will depend on how well the data can be described by a mixture of Gaussians, and the deviations could potentially be used to guide construction of a better statistical model.

3.7. Data Importance

Rather than being interested in what features contribute to a model, we may be interested in what data contributes to a model. For linear models, we may talk about points with high 'leverage', for example. The series expansion method can be adapted for calculating the influence of individual or sets of data points. The idea is broadly the same: we can retrain the model by removing a subset of data points and comparing it against the original model. The models can be compared locally or globally in analogous fashion to equations 1, 2 and 3. It may also be of interest to replace the data point with a resampled point to investigate the model stability across multiple experiments.

3.8. Relationship to Adversarial Perturbations

There has recently been interest in identifying, suppressing, and understanding adversarial examples. Adversarial examples 'look like' normal data points, but cause statistical models to produce spurious predictions. The process of creating adversarial examples tends to focus on changing the value of one or more features. Another kind of adversarial transformation, that to the best of our knowledge has not been described in the literature, could be to adversarially censor the value of one or more features. As when identifying feature importance, the adversarial examples will be sensitive to the loss function, which in turn is determined by the context.

For example, the optimal censoring attack involves removing the feature that has the greatest effect on the local loss,

$$\Delta L = L(y, \delta(x)) - L(y, \delta(x \backslash x_i)), \tag{68}$$

where the backslash indicates the omission of a feature value.

Likewise, an attacker may want to 'poison' the training data set, which would correspond to removing the most important data points, as described above. These points can be identified using the terms in our power series.

3.9. Utility of the Power Series Formulation

In this section, we have analysed the properties of the power series formulation in greater detail. We have shown the power series formulation has a number of pleasing properties. It is invariant under monotonic transformations. Informally, this says that the power series is insensitive to the units used for representing a feature. If we were to represent a length using centimetres, inches, or on a logarithmic scale, the power series would still give the same answer. There is also a simple formula for relating the feature importance of compound features to the feature importance of its constituents, which may be of interest when trying to cluster features together.

Feature importance was visualized for a simple example. The distribution of feature importance values helped to elucidate that even for the optimal model, additional information could harm the model performance for a subset of data points, and we were able to relate these instances to the underlying distribution. This demonstrates potential for understanding more complex data sets. While we mainly applied the power series formulation to features, it has wider applicability, such as understanding how data points contribute to the training of a model.

4. CONNECTION TO SHAPLEY VALUES

The power series formulation has a parallel in game theory called Shapley values [15]. Shapley values were conceived as a way of fairly distributing output among a group of players. Moreover, they are the only way of distributing an output that satisfies symmetry, consistency, efficiency, and linearity [16]. These are considered intuitively reasonable properties and can be translated into equivalent statements for feature importance [17]. Establishing the formal relationship between our power series formulation and Shapley values improves our method's credibility and reinforces its theoretical foundations.

4.1. Defining Shapley Values

In Shapley's game, each player has the option of joining a coalition to produce a scalar output. The output will depend on the coalition's composition. Combinations of players may act synergistically to create a total output that is larger than the sum of the individual contributions, or conversely their contributions may be subadditive. A fair distribution will separate the output in proportion to each player's marginal contribution. However, owing to the non-linearity, this will be affected by the order in which players join the coalition.

Shapley overcame this problem by calculating the marginal contribution averaged over every possible ordering. It can be calculated using the formula

$$S_i = \sum_{r \subseteq q \setminus \{i\}} \frac{|r|!(|q| - |r| - 1)!}{|q|!} [h_{r \cup \{i\}}(x_{r \cup \{i\}}) - h_r(x_r)], \tag{69}$$

where S_i is the Shapley value of player i, q is the set of all players, r is a subset of players not including player i, h is the function that calculates the coalition's output, x indicates a set of features, and the sub-scripts of h indicate which players are involved in calculating the coalition's output.

4.2. The Relationship Between Shapley Values and Feature Importance

Shapley values can be adapted for calculating feature importance [17, 18]. The set of players is replaced by a set of possible feature values, and the function h is replaced by a measure of model performance, typically accuracy. When calculating standard Shapley values, the features are fixed, but can be present or absent. The Shapley value for each feature is therefore a scalar value, S_i. When treating feature importance for statistical models, the features can typically take multiple values, so the Shapley values need to be replaced with a function, $S_i(x)$. The local nature of the

Shapley feature importance is often treated implicitly in the literature, which may lead to confusion. A global measure of feature importance ($S_i(X)$) can be derived by integrating over the feature distribution [19, 20], but is rarely used. Likewise, we could derive an instance feature importance ($S_i(x, y)$) by using the loss function.

Shapley feature importance has a peculiar interpretation: $S_i(x)$ measures how a particular feature, x_i, contributes to a model's accuracy against a naïve baseline [21]. It does not measure the change in accuracy when a feature is removed, nor does it provide a measure of model sensitivity to changes in that feature. These relationships can be captured through other measures that we describe in Section 5.

4.3. Equivalence of Shapley Values and the Power Series Formulation

Our power series representation relates feature importance to how the model's loss changes in response to the inclusion of one or more features (equation 9), while Shapley values are derived from the average marginal contribution across all possible permutations of features (equation 69). Despite their ostensibly different structures, a correspondence can be formally established between them using combinatorics.

We can use symmetry arguments to motivate the correspondence: if there is an interaction between feature i and j, then i will occur later than j in half of all the permutations and it will receive half of the interaction term V_{ij}. Similarly, i will receive a third of any cubic interaction and so forth. The Shapley values receive an equal share of each interaction term, so they can be related to the power series by

$$S_i = V_i + \sum_{i,j} \frac{1}{2} V_{ij} + \frac{1}{3} \sum_{i,j,k} V_{ijk} + \dots \tag{70}$$

where we define terms with repeat indices as zero.

We sketch the formal combinatorics to prove equation 70. We do this by showing each interaction term contributions proportionality to each Shapley value. Without loss of generality, consider the interaction between feature 0 and the next $p - 1$ features,

$$V_{0 \dots p}, \tag{71}$$

and its contribution to the Shapley value of feature 0, S_0.

We break the calculation into two stages: first, we consider all permutations of features that have feature 0 in the m-th position (using 0 indexing) and the group of $p-1$ features that interact with feature 0 preceding it - this provides an interaction term between feature 0 and the $p-1$ other features. Second, we sum over all possible positions of feature i to calculate the total contribution. m cannot occur before the $p-1$-th position, because otherwise it wouldn't be possible to have all of the other $p-1$ terms interacting with it, so we only use values of m between $p-1$ and $n-1$, inclusive. Following these steps, we show the Shapley value is equivalent to a weighted sum of our power series terms.

If feature 0 is in position m, then there are $n-p$ remaining features that can be arranged in $m-p+1$ places. Accounting for the possible orderings means the total number of permutations is

$$\binom{n-p}{m-p+1} m!(n-m-1)! \tag{72}$$

Next, we need to sum over all possible positions m. Again, the possible values of m are restricted, so the total number of permutations is

$$\sum_{m=p-1}^{n-1} \binom{n-p}{m-p+1} m!(n-m-1)! \tag{73}$$

We need to divide by $n!$ to account for the number of possible permutations, so the contribution of $V_{0\ldots p}$ to S_0 is

$$\frac{1}{n!} \sum_{m=p-1}^{n-1} \binom{n-p}{m-p+1} m!(n-m-1)! V_{0\ldots p} \tag{74}$$

After a straight-forward but tedious calculation this reduces to

$$\frac{1}{p} V_{0\ldots p}. \tag{75}$$

Summing over every interaction term involving feature 0 we get

$$S_0 = V_0 + \sum_{0,j} \frac{1}{2} V_{0j} + \frac{1}{3} \sum_{0,j,k} V_{0jk} + \ldots \tag{76}$$

We have recovered the Shapley value for S_0, demonstrating the equivalence in equation 70.

The base rate is not incorporated into any of the Shapley values, although it will affect the model performance, and may be relevant to their interpretation in terms of what information is available.

4.4. Calculating Shapley Values

In section 2.7, we examined a linear model with features drawn from a correlated, bi-Gaussian distribution. This serves as a useful example for calculating Shapley values since there is a nice, closed-form solution. We will only calculate the importance of x_0 for brevity, but x_1 follows an almost identical procedure. First, we generate the possible permutations of features. These are simply

$$[x_0, x_1] \tag{77}$$

and

$$[x_1, x_0]. \tag{78}$$

The terms for $h_{r\cup\{i\}}(x_{r\cup\{i\}}) - h_r(x_r)$ in the Shapley calculation are

$$E_{X,Y}[L(y, \delta(x_0))] - E_{\delta|\emptyset}[L] \tag{79}$$

and

$$E_{X,Y}[L(y, \delta(x_0, x_1))] - E_{X,Y}[L(y, \delta(x_1))]. \tag{80}$$

Using the values we calculated before (and setting σ = 0 to reduce clutter),

$$E_{X,Y}[L(y, \delta(x_0))] - E_{\delta|\emptyset}[L] = \beta^2(x_1 - \rho\frac{\sigma_1}{\sigma_0}x_0)^2 - (\alpha x_0 + \beta x_1)^2, \tag{81}$$

and

$$E_{X,Y}[L(y,\delta(x_0,x_1))] - E_{X,Y}[L(y,\delta(x_1))] = 0 - \alpha^2(x_0 - \rho\frac{\sigma_0}{\sigma_1}x_1)^2. \quad (82)$$

The Shapley value for x_0 is

$$S_{x_0} = \frac{1}{2}E_{X,Y}[L(y,\delta(x_0))] - E_{\delta|\emptyset}[L]] + \frac{1}{2}E_{X,Y}[L(y,\delta(x_0,x_1))] - E_{X,Y}[L(y,\delta(x_1))] \quad (83)$$

$$= \frac{1}{2}(\beta^2(x_1 - \rho\frac{\sigma_1}{\sigma_0}x_0)^2 - \alpha^2(x_0 - \rho\frac{\sigma_0}{\sigma_1}x_1)^2 - (\alpha x_0 + \beta x_1)^2). \quad (84)$$

Alternatively, we note that equations 79 and 80 are equivalent to the first-order and second-order interactions from section 2.7, so the Shapley value can be written as

$$S_{x_0} = v_0(x_0,x_1) + \frac{1}{2}v_{01}(x_0,x_1) \quad (85)$$

$$= \beta^2(x_1-\rho\frac{\sigma_1}{\sigma_0}x_0)^2-(\alpha x_0+\beta x_1)^2+\frac{1}{2}((\alpha x_0+\beta x_1)^2-\beta^2(x_1-\rho\frac{\sigma_1}{\sigma_0}x_0)^2-\alpha^2(x_0-\rho\frac{\sigma_0}{\sigma_1}x_1)^2) \quad (86)$$

$$= \frac{1}{2}(\beta^2(x_1 - \rho\frac{\sigma_1}{\sigma_0}x_0)^2 - \alpha^2(x_0 - \rho\frac{\sigma_0}{\sigma_1}x_1)^2 - (\alpha x_0 + \beta x_1)^2), \quad (87)$$

as above.

For small examples, both approaches require a similar amount of effort. However, in the next section we show that sometimes the power series can yield much faster calculations.

4.5. Faster Calculation of Shapley Values

Shapley values are calculated by averaging over $n!$ possible permutations, each of length $n+1$ when including the no feature case. The total number of models to evaluate for the Shapley values calculation is then $(n+1)n!$. This can be reduced by reusing some of the sub-sequences in

the permutations [22]. Nevertheless, the number of evaluations required quickly becomes impractical.

We can reduce the computation considerably using the power series formulation if we have prior knowledge about the values of coefficients or there are structural constraints, such as when interactions only occur between pairs of features. We can form a single permutation and calculate the feature importance from the marginal change in performance between adjacent terms, for a total of only $n + 1$ calculations. Similarly, if only first and second-order terms are present, we only need to sample n permutations, for a total of n^2+1 evaluations. We demonstrate this by construction below.

To calculate the total number of permutations we first calculate the empty model and then sample a random permutation. We will assume for simplicity (and without loss of generality) that the permutation is

$$\pi_0 = (f_0, f_{n-1}, f_{n-2}, \dots, f_1), \tag{88}$$

where f_i represents the addition of feature i to the set of features used to construct the statistical model. The set of features is sequential, so the second entry in π includes features f_0 and f_{n-1}. If we are calculating global feature importance, these terms would explicitly be

$$\pi_0 = (E_{X,Y}[L(y, \delta(x_0))], E_{X,Y}[L(y, \delta(x_0, x_{n-1}))], \dots, E_{X,Y}[L(y, \delta(x_0, \dots, x_{n-1}))]). \tag{89}$$

We can then shift the permutation right, up to $n - 1$ times, and re-evaluate the model performance,

$$\pi_1 = (f_1, f_0, f_{n-1}, f_{n-2}, \dots, f_2), \tag{90}$$

…

$$\pi_{n-1} = (f_{n-1}, f_{n-2}, \dots, f_1, f_0). \tag{91}$$

The diagonal entries all involve f_0 and interactions with other features. The diagonal pattern is replicated for the other features, too. The structure allows the individual and second-order interactions to be easily deduced. The first entry in each permutation provides the individual feature

contributions. Assuming we have calculate $E_{X,Y}[L(y, \delta(\emptyset))]$ once, the first-order terms are

$$V_0 = E_{X,Y}[L(y, \delta(x_0))] - E_{X,Y}[L(y, \delta(\emptyset))] \tag{92}$$

and so.

The second entry in each permutation is the combination of two known individual contributions (for example, V_0 and V_{n-1}), and one unknown second-order interaction (for example $V_{0,n-1}$). Basic arithmetic allows the second-order interaction to be identified. Using the first two terms in π_0 and the first term in π_{n-1},

$$V_{0,n-1} = E_{X,Y}[L(y, \delta(x_0, x_{n-1}))] - E_{X,Y}[L(y, \delta(x_0))] - E_{X,Y}[L(y, \delta(x_{n-1}))] - E_{X,Y}[L(y, \delta(\emptyset))], \tag{93}$$

or

$$V_{0,n-1} = \pi_0[1] - \pi_0[0] - \pi_{n-1}[0] + E_{X,Y}[L(y, \delta(\emptyset))], \tag{94}$$

where we are using the square brackets to identify entries in the permutation (note that the last term changes sign because the permutation terms each include an empty set contribution.)

As we have assumed there are no higher-order interactions, a similar process follows for the third entry in each permutation and so on. Iterating this process allows all the interaction terms to be determined from only $n^2 + 1$ evaluations. Shapley values can be immediately derived from the power series if desired.

We can reduce the number of evaluations even further by noting that we don't have to calculate permutations. We can generate a model using any subset of features without having to generate any intermediate models. This allows us to map the inclusion of features onto a 2-level design problem, and use the design of experiments to minimize the number of required evaluations [23]. For example, fractional factorial designs can suppress 'aliasing', the confounding of low and high-order interactions. These procedures allow Shapley values to be calculated even for complex models including those with a large number of dimensions.

4.6. Analogous Properties

Shapley values are renowned for having several desirable properties: efficiency, symmetry, linearity, and null player. These each have close analogs in the power series formulation (as may be expected). We briefly describe and prove each.

Efficiency says the sum of all the terms in the series should equal the expected loss,

$$E_{Y|x}[L(y,\delta(x))] = \sum_{k\subseteq\{0,\dots,n-1\}} v_k(x). \tag{95}$$

This is true by construction.

Symmetry says that if two features (x_i and x_j) change the expected loss in an identical manner, then they must have identical power series coefficients,

$$E_{Y|x}[L(y,\delta(x_s \cup x_i))] = E_{Y|x}[L(y,\delta(x_s \cup x_j))]\ \forall\ x_s \in x \setminus x_i, x_j \Rightarrow v_{i,s} = v_{j,s}. \tag{96}$$

Since the expected losses are the same we have

$$v_{S\cup i}(x) = E_{Y|x_{S\cup i}}[L(y,\delta(x_S \cup x_i))] - \sum_{s\subsetneq S} v_s(x \cup x_i) \tag{97}$$

$$= E_{Y|x_{S\cup j}}[L(y,\delta(x_S \cup x_j))] - \sum_{s\subsetneq S} v_s(x \cup x_j) = v_{S\cup j}(x). \tag{98}$$

Therefore

$$v_{i,s}(x) = v_{j,s}\ \forall\ s \in \{0,\dots,n-1\} \setminus \{i,j\}. \tag{99}$$

Linearity says that if the loss can be decomposed into two other loss functions (L_0 and L_1), then the power series coefficients for L are also a linear combination of the power series coefficients for L_0 and L_1,

$$L = \alpha L_0 + \beta L_1 \Rightarrow v_i = \alpha v_i^0 + \beta v_i^1, \tag{100}$$

where the superscripts indicate which loss function the coefficients belong to. This follows from the linearity of expected values,

$$v_\emptyset + \sum_{i=0}^{n-1} v_i \cdot I_i + \sum_{i=1}^{n-1}\sum_{j=0}^{i-1} v_{ij} \cdot I_i \cdot I_j + \sum_{i=2}^{n-1}\sum_{j=1}^{i-1}\sum_{k=0}^{j-1} v_{ijk} \cdot I_i \cdot I_j \cdot I_k \ldots \tag{101}$$

$$= E_{Y|x}[L(y, \delta(x))] = E_{Y|x}[(\alpha L_0(y, \delta(x)) + \beta L_1(y, \delta(x)))] \tag{102}$$

$$= \alpha E_{Y|x}[L_0(y, \delta(x))] + \beta E_{Y|x}[L_1(y, \delta(x))]. \tag{103}$$

$$= \alpha(v_\emptyset^0 + \sum_{i=0}^{n-1} v_i^0 + \cdots) + \beta(v_\emptyset^1 + \sum_{i=0}^{n-1} v_i^1 + \cdots). \tag{104}$$

Therefore

$$v_i = \alpha v_i^0 + \beta v_i^1 \; \forall \; i \subseteq \{0, \ldots, n-1\} \tag{105}$$

Null player says that a feature x_i that provides no predictive power should have all terms in the power series zero. Since

$$E_{Y|x}[L(y, \delta(x_s \cup x_i))] = E_{Y|x}[L(y, \delta(x_s))] \; \forall \; x_s \in x \setminus x_i, \tag{106}$$

it follows from equation 12 that all power series terms involving i are zero,

$$v_{i,s} = 0. \tag{107}$$

Similar proofs follow for the instance and global feature importance by substituting ν or V for v, and either replacing the expectation with a single value of y or using the expectation $E_{X,Y}$.

4.7. Contrasting Power Series and Shapley Values

Shapley values do not explicitly indicate how features interact together. There can be ambiguity when features contribute super- or subadditively. The ambiguity can be resolved by moving to the more expressive power series formulation. This is demonstrated through three probability distributions with different interactions between features, but identical Shapley values. Each distribution has two binary features (x_0 and x_1), a single class (y = 0 or 1), and accuracy as the performance metric. These distributions are shown in Figure 2.

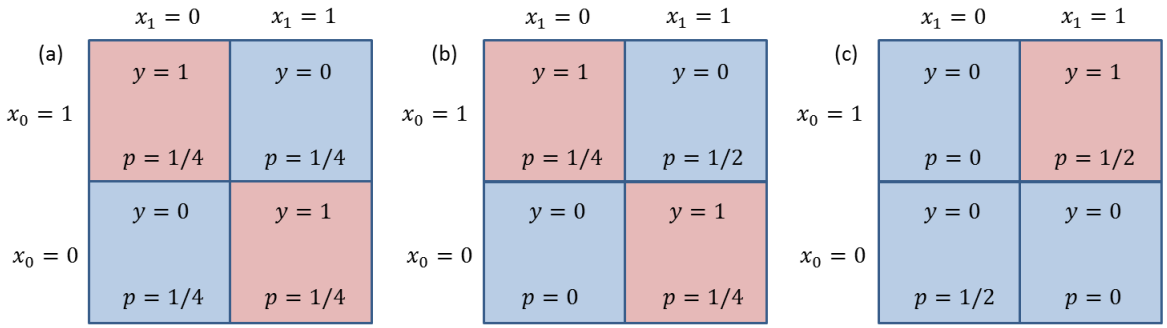
The first distribution is an 'exclusive-OR' problem with uniform and uncorrelated probabilities for each feature, the second distribution has some correlation between the two features, and the third distribution has strong redundancy - all three features are perfectly correlated (x_0 = x_1 = y) with $p(y = 0) = 1/2$.

In the first instance, both features need to be observed for there to be any gain in predictive accuracy. The power series expansion indicates this through individual contributions of 0 and a positive interaction term of 1/2, the second distribution has individual contributions of 1/4 each and no interaction terms, and the third distribution has two individual contributions of 1/2 and a negative interaction of -1/2. These situations are ambiguous when Shapley values are used, while the contributions of each term are clear using the power series formulation.

While the power series formulation provides a more nuanced view of feature interactions, this comes at the cost of some added complexity. The greater number of feature importance terms can make the interpretation and analysis more complicated than for standard Shapley values. In some cases, the complexity may also increase the computational difficultly. The power series formulation and Shapley values both require 2^n permutations to be calculated for exact solutions. However, a number of approximations have been developed for calculating Shapley values and it is unclear if these can be immediately applied to the power series formulation [24–26]. Additionally, there is usually no closed-form solution for directly calculating the feature importance for complex models, so the terms need to be estimated numerically, often through some form of Monte Carlo simulation. The greater number of terms in the power series formulation means that convergence of these numerical estimates may be slower than for a direct calculation of the Shapley values, although we have not attempted a numerical comparison of existing approaches.

4.8. Shapley Values for Compound Features

We defined power series terms for compound features in section 3.3. Following a similar procedure we can define Shapley values for compound features. The literature contains several inconsistent definitions for feature interactions [9]. Feature interactions would form a natural basis for defining compound Shapley values. However, it is unclear which, if any, definition is best. We will



	Expected accuracy		
No features	$1/2$	$1/2$	$1/2$
x_0	$1/2$	$3/4$	1
x_0, x_1	1	1	1

Figure 2 Three sampling distributions with identical Shapley distributions, but different probability structures. The probability density function is represented by boxes along the top row and the expected accuracy of the Bayes-optimal model is given as function of available features along the bottom row. a) An exclusive-OR function with uniform probabilities, b) An exclusive-OR function with non-uniform probabilities, c) An AND-function with the states $x_0 = x_1 = 0$ and $x_0 = x_1 = 1$ having equal probabilities.

define the compound feature Shapley values by using the standard formula for Shapley values and replacing a single feature with a set of features. Following our approach for the power series (section 3.3), we will introduce the compound feature using a pair of features, $\{i, j\}$,

$$S_{\{i,j\}} = \sum_{r \subseteq q \setminus \{i,j\}} \frac{|r|!(|q| - |r| - 1)!}{|q|!} [h_{r \cup \{i,j\}}(x_{r \cup \{i,j\}}) - h_r(x_r)], \tag{108}$$

where the features are defined in section 4.1, and these can be 'translated' into our formalism. Following considerations of symmetry, we deduce that the Shapley value will decompose to

$$\begin{aligned} S_{\{i,j\}} = (V_i + V_j + V_{ij}) &+ \frac{1}{2} \sum_{k \subseteq \{0,\ldots,n-1\} \setminus \{i,j\}, k \neq \emptyset} (V_{ik} + V_{jk} + V_{ijk}) \\ &+ \frac{1}{3} \sum_{k,l \subseteq \{0,\ldots,n-1\} \setminus \{i,j\}, k,l \neq \emptyset} (V_{ikl} + V_{jkl} + V_{ijkl}) + \ldots \end{aligned} \tag{109}$$

where $\{0, \ldots, n-1\}$ represents the set of possible features and k is used as a dummy index for the summation. Generalizing to compound features involving any set of features Ω,

$$S_{\{\Omega\}} = \sum_{\omega \subseteq \Omega} \left[V_\omega + \frac{1}{2} \sum_{k \subseteq \{0,\ldots,n-1\} \setminus \Omega, k \neq \emptyset} V_{\omega k} + \frac{1}{3} \sum_{k,l \subseteq \{0,\ldots,n-1\} \setminus \Omega, k,l \neq \emptyset} V_{\omega kl} + \ldots \right] \tag{110}$$

The Shapley values from equation 110 match those from the modified power series given in equation 62, as expected.

It is interesting to note that the compound Shapley values are not the sum of their constituent Shapley values, that is,

$$S_{\{i,j\}} \neq S_i + S_j. \tag{111}$$

Since the sum of the Shapley values is fixed, this implies that regrouping features together alters the Shapley values of the untransformed features, too. To demonstrate this effect concretely, consider a class y that is determined by the parity of three features, x_0, x_1, x_2, which are independently distributed with equal probabilities for their two states. If the three features are treated separately each will have the same Shapley value of $1/6$ (the change in predictive accuracy from 1/2 to 1, divided by three). If two of the features are combined into a compound feature, then both the remaining features will have Shapley values of $1/4$. The synthesis of features is therefore non-additive and affects features in a non-local way. This is a consequence of the symmetry property that underlies Shapley values (see Fujimoto *et al* [9] for a list of Shapley properties). It may be possible to create an additive version of Shapley values. For the power series approach this might be through a weighting factor, or for the standard definition of Shapley values this could possibly be accomplished by weighting the orderings of the permutations [20, 27]. We leave this as an open problem.

5. ALTERNATIVE MEASURES OF FEATURE IMPORTANCE

There are many formulations of feature importance. Identifying the core principles involved in each formulation can help to clarify when each is valid, and when they can be expected to provide consistent conclusions. These methods were chosen for their diversity and popularity, and are non-exhaustive. Nevertheless, they provide contrasting perspectives on what principles can be used to define feature importance and their applications to different problems.

We compare our power series expansion against a number of other methods in Table 2 and through the discussion below. The columns in Table 2 attempt to encapsulate, somewhat imperfectly, the main attributes of each method. We will briefly unpack each of these attributes.

Metric refers to how the model performance is quantified. Only two were identified in the measures of feature importance we considered: 'Predictive' that quantifies how closely a predicted and observed output match, such as accuracy, and 'fidelity' that quantifies the similarity between the model outputs before and after the features have been perturbed. Other possible metrics include model size and computation time.

Principle captures the core idea used by each method for determining feature importance. Some of the most common principles encountered in the literature are:

- **Permutation** The importance of a feature is determined by comparing the model with and without randomly permuting the feature value between data points. This has a similar effect to randomly resampling some of the features independently of the data point's dependent feature. The principle could be generalized by introducing other kinds of randomization, such as Gaussian noise, although to the best of our knowledge this has not been used for calculating feature importance to date.

- **Imputation** Feature importance is measured by measuring changes in model performance when missing features are imputed using the conditional probability density or with a function that generates a 'best guess' for the missing features using the known features.

- **Retraining** The original model is compared with a separate model trained on a data set with some features removed.

- **Sensitivity analysis** The change in a model's output is measured in response to a perturbed input.

- **Model-specific** There are some measures of feature importance that are specific to certain models or representations, called model-specific measures. These are usually difficult

to directly compare against model-agnostic methods and may have little in common with each other except for their limited applicability.

- **Data point influence** Some data points may have a disproportionate effect on what model is learnt. Data point influence is measured using similar principles to those found in feature importance methods, except data points are perturbed or removed, rather than features.

- **Unsupervised learning** These methods try to find structure in the model or data. Unlike feature importance methods, they do not quantify model performance. They provide a different approach to understanding model structure and can be used in conjunction with feature importance methods.

The **Scope** of the method is 'local' if it operates on a single point or neighbourhood, or 'global' if it addresses the performance of the entire model.

The **Output** can be unique or plural. A unique output indicates there is a single measure of feature importance that is self-consistent and coherent. We recognize the output as unique even if there are multiple values, so long as they provide a single picture. Thus we consider a collection of Shapley values unique. We use plural to indicate that there may be inconsistent or independent measures from a single formulation. For example, it is possible to generate multiple anchors or counterfactuals for a data point, and these may provide conflicting views about which features are important.

Free parameters identifies if the model has tuning parameters that can be specified by the user, but exclude choices that are related to the quantity the method calculates. For example, we identify surrogate models as having a free parameter because the user can arbitrarily choose a type of model, such as decision tree or linear regression, and can impose other constraints like the maximum depth of a tree. While we exclude the choice of loss function for our method it is dictated by the context or scenario, and should not be arbitrarily chosen by the user. The distinction of what constitutes a free parameter in this context can sometimes be ambiguous or subjective.

5.1. Partial Dependence Plots

Partial dependence plots calculate the average dependence of a model on a subset of features x_s and average over the remaining complement of features, x_c, that are treated as independent [28]. Formally the definition of the partial dependence is

$$\delta_s(x_s) = \int \delta(x_s, x_c) p(x_c = X_c) dX_c, \tag{112}$$

Table 2 Comparison of high-level attributes of methods of determining feature importance. Details provided in text.

Method	Metric	Principle	Scope	Output	Free parameters
Partial dependence	N/A	Permutation	Local	Unique	No
Permutation importance	Predictive	Permutation	Global	Unique	No
Sensitivity analysis	Predictive	Sensitivity analysis	Local	Plural	Yes
Anchors	Fidelity	Sensitivity analysis	Local	Plural	Yes
Surrogate models	Fidelity	Retrain on model data	Local or Global	Plural	Yes
Node importance	Predictive	Model-specific	Global	Plural	No
Influential data points	Predictive	Retrain on data subset	Local or Global	Unique	No
Representative data points	N/A	Unsupervised learning	Global	Plural	Yes
Coordinate transformation	N/A	Unsupervised learning	Local or Global	Unique	Sometimes
Shapley additive explanations	Fidelity	Impute data assuming independence	Local	Unique	No
Power series	Predictive or Fidelity	Various	Local or Global	Unique	No

where the terms have a similar meaning to that use in our formalism: δ is a statistical model that predict y using x, x_s are the subset of features we are plotting, x_c is the complement, and δ_s is the reduced model that only uses a subset of features. The partial dependence plot visualises the model's structure and is usually applied to features with continuous values. It does not numerically quantify feature importance and is independent of the model's predictive performance. Individual Conditional Expectation plots [29] and Accumulated Local Effects [30] rely on similar principles.

Our Bayesian formulation of missing data accounts for the correlation between the subset and complement by averaging over the conditional probability of the complement, so it is effectively

$$\delta_s(x_s) = \int \delta(x_s, x_c) p(x_c = X_c | x_s) dX_c. \tag{113}$$

This alternative formulation was suggested by Friedman [28], although he cautioned against its use because he was concerned it would confound the influences of x_s and x_c.

5.2. Permutation Importance

The permutation test is a classical non-parametric method for checking for trends or associations between features [31]. It has since been adapted for high-dimensional statistical models [32]. The model performance is evaluated before and after permuting the values of a feature, either at training or test time. The relative change in performance for each permuted feature provides a measure of feature importance.

Permuting the feature at training time is similar to removing the feature from the model entirely, and will roughly correspond to removing all first and higher-order power series terms associated with that feature. Applying the permutation to the test data is similar to the partial dependence plot with a single feature in the complement of equation 112.

The permutation importance does not explicitly separate first and higher-order terms, and there can be ambiguity about whether features contribute individually or as a synergistic effect. If there are three perfectly correlated binary features and the model uses a majority vote to determine the class, then the permutation test will show that none of the features are important, which is clearly not true.

Permuting the features is similar to introducing noise, so can be thought of as measuring model robustness. This is different to hiding information, and there may be disagreement between our power series formulation and the results of a permutation test.

5.3. Counterfactuals

Counterfactuals are a form of sensitivity analysis. They identify what minimal change in the input would be required for the statistical model to make a different prediction or recommendation, where the size of the change is quantified by a chosen metric [33]. In general there will be multiple counterfactuals. For example, a counterfactual for a home loan could quantify how much an applicant's income would need to increase for the loan to be approved. Another counterfactual for the same applicant could identify what change in credit score would allow the loan to be approved. Some of the counterfactuals may even produce combinations of features that are implausible or impossible. The user may choose to favour one counterfactual based on contextual relevance, although in general there will be ambiguity about which counterfactual, if any, is the most "interpretable".

Counterfactuals are a fundamentally different way to the power series formulation for looking at feature importance. They examine model sensitivity to local feature variation and do not consider the impact of missing data. Counterfactuals are arguably better suited to situations in which one in trying to steer a recommendation or decision, such as a loan application, while the power series is better placed to address missing data or the costs of data collection.

5.4. Anchors

Anchors are a subset of features that, within a local neighbourhood, are sufficient to probabilistically guarantee a model's output [34]. There is a trade-off between the probability that the guarantee is met and the coverage of the rule. Generally, localized rules will provide outputs with high confidence, but are rarely applicable, and vice versa for larger neighbourhoods. The interpretation of anchors is that the subset elucidates the locally important features. Similar to the counterfactuals, there can be ambiguity in the interpretation of features since the anchors are usually non-unique. When anchors are deterministic, there is a one-to-one correspondence between anchors and counterfactuals [35], and they share many properties.

Deterministic anchors can be identified from the power series formulation. These will be local measures of feature importance that exclude all first and higher-order terms associated with a feature. Anchors are defined for a local neighbourhood, while our method uses point or global estimation. Our method could be modified to sample only around a pre-defined neighbourhood to increase the similarity to anchors.

5.5. Surrogate models

There is sometimes value in training one model, called a surrogate model, that mimics the behaviour of a different model [36]. This may be of interest because the original model is expensive in

terms of computation cost or memory footprint [37], to increase model robustness [38], or perhaps the model is proprietary and we cannot access its internal structure [39, 40]. A surrogate model can potentially achieve similar performance without these downsides.

Surrogate models are also used to transform complex models into simpler, more interpretable ones. These surrogate models are generally shallow decision trees or linear regressions [3]. The surrogate model can be trained to mimic the other model's structure for a local region [14] or globally [36]. The user can then use the structure of the surrogate model to understand the behaviour of the original, more complicated model.

Surrogate models are agnostic with respect to the original model, and there is freedom for the end user to shape the surrogate model to highlight features of interest. There can often be multiple surrogate models that perform well, creating ambiguity in which 'interpretable' surrogate model is the correct one [3]. Other issues are that the surrogate model may perform poorly in areas of low sample density, that they will provide correct behaviour for only some data points (otherwise they would be as uninterpretable as the original model), and as we showed in Section 2.6, highly faithful models can suffer from degraded performance.

When maximizing the fidelity, our approach can be thought of as creating a global surrogate model with missing features.

5.6. Node Importance for Decision Trees

There are several measures of feature importance that are calculated from a model's parameters or structure. These techniques cannot be applied to other representations or types of models. A popular example is a measure of feature importance for decision trees proposed by Breiman [32]. The importance measure is generated from changes of the squared error risk at the internal nodes of a decision tree.

There are usually several decision trees that can be constructed for a single mapping, as shown in Figure 3, which means the feature importance is not usually uniquely defined. The lack of uniqueness affects some other model-specific measures. Neural networks can be visualized by identifying images that maximize a neurons activation [41], but it can be possible to find two neural networks with different connections that nevertheless produce identical predictions. We suggest as a general principle that feature importance should be independent of the model representation, since the representation can be modified with no change in predictive performance.

Our method is independent of a model's internal representation and cannot be meaningfully compared against node importance or other model-specific measures. We suggest that dependence on the internal structure of the model in measures of feature importance is best avoided if pos-

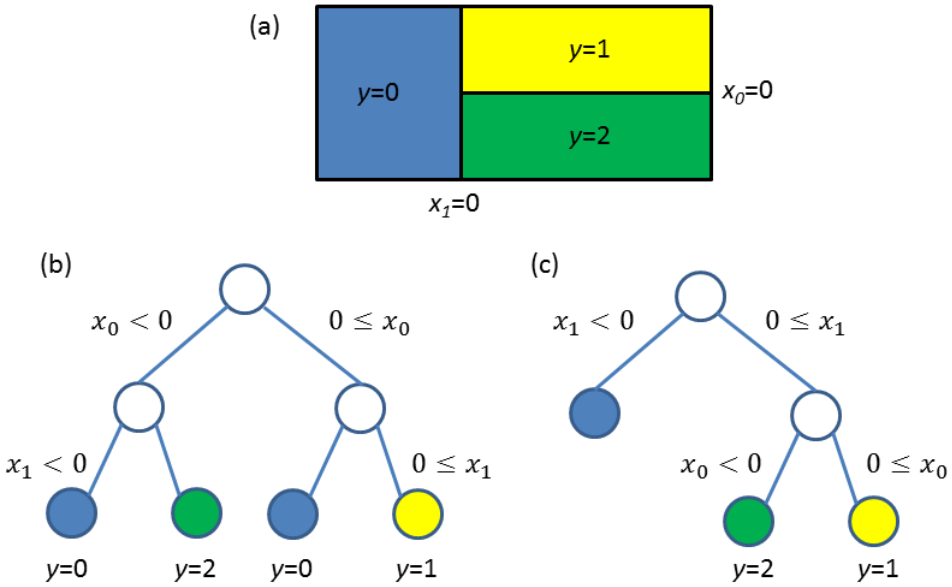


Figure 3 The mapping of features to prediction does not uniquely define a decision tree's internal structure. a) A simple mapping from two features to three possible classes. A possible decision tree representation that first splits on the feature x_0 (b) and x_1 (c).

sible, otherwise the feature importance is not uniquely defined.

5.7. Influential Data Points

While data point sensitivity is distinct from feature importance, we include it here because of the similarities between the methods. Data point sensitivity has classically been investigated with the bootstrap, jackknife and other methods of data resampling [42]. More modern methods usually rely on similar principles. Commonly, a single data point is removed from the training set and a new model is constructed [43]. Approximations are often necessary to avoid the severe computation time of retraining the model multiple times [44, 45]. Our power series formulation can work cooperatively with these approximations to quickly generate detailed views of how data points contribute to the final model.

5.8. Representative Data Points

It can be useful to visualize a small subset of data points that highlight key properties of the probability distribution. There are a huge number of clustering and outlier detection methods in the literature. One example is prototypes and criticisms developed by Kim *et al* [46]. Prototypes represent typical points in the distribution and exemplify classes. Criticisms are atypical data points designed to highlight the complexities of the sample space. Visualization of data points is a local process and provides a good counterbalance against our measure of feature importance that aver-

ages over the sampling distribution, and may therefore be insensitive to some outliers.

5.9. Coordinate Transformations

Features can become more interpretable following a coordinate transformation. There are classical methods like principal component analysis, and more sophisticated kernel and stochastic decomposition methods. We view these transformations as complementary to the power series method. Coordinate transforms can be applied prior to training the model or during testing to concentrate the importance into a few key features (see section 3 for some discussion about feature transformation).

5.10. Shapley Additive Explanations

Shapley additive explanations (SHAP) [17] are closely related to our approach. The SHAP explanation model is generated by mapping the original model that uses all of the data to a smaller model with missing values. Feature independence and model linearity are assumed, while our approach uses the conditional probability distribution. The SHAP model decomposes the feature contributions into a set of Shapley values and uses kernels to improve feature interpretability. Our power series formulation decomposes Shapley values into interaction terms. We briefly discussed the use of kernels or other transformations (see Section 3.3), but have not analysed their application in detail. A global variant of SHAP, Shapley Additive Global Importance (SAGE), was recently proposed [47].

5.11. Comparison with Other Shapley Formulations

Numerous methods, like SHAP, have been proposed for modifying Shapley values to calculate feature importance. The unifying ingredient of these methods is that they each use a modified version of Shapley's formula, given in Equation 69, to calculate the average marginal contribution of each feature. The methods differ in their assumptions, their treatment of missing features, and their implementation details. These differences are non-trivial, and can produce different numerical outputs. In this sub-section we discuss some ways in which the methods vary and compare them with our power series formulation. These relate to the choices we made when defining our formulation of feature importance in Section 2.

- **Metric.** Most formulations assume accuracy is the key metric of performance [7, 10, 22] Goodness-of-fit (R^2) has sometimes been used for linear models [48]. Our formulation starts with a generic loss function L, and we examined the specific cases of accuracy and fidelity in section 2.6.

- **Model-dependence.** Feature importance can be defined intrinsically as a correlation between independent and dependent features for a sampling distribution, or extrinsically by the impact a feature has on a particular model. Most formulations can only be used to describe the extrinsic feature importance (see [18] and references therein), while our formulation allows both intrinsic and extrinsic feature importance to be treated.

  A second way model-dependence can arise is through coupling between the model's internal representation and the feature importance. This is a common property of feature importance measures developed for neural networks, for example, see [17, 24, 49]. Many of the formulations in the literature are specific to certain learning algorithms like linear models or artificial neural networks. These properties reduce the applicability of the methods and often violate the principle we suggested in section 5.6 that feature importance should only be sensitive to the model's output. Our formulation is applicable to any standard statistical model and can be adapted for other scenarios, like data importance (see section 3.7).

- **Baseline for comparison.** The treatment of missing features varies throughout the literature. Some methods try to impute the missing values using a single baseline value, like the mean, median, or mode [18]. Others impute a distribution for the missing values that can be dependent [50] or independent [51] of the known feature values. Yet another method is to retrain the model with a subset of features removed [51]. These approaches for treating missing data are not equivalent and will produce different results. We suggest model retraining is appropriate for measuring feature importance for a learning algorithm, while imputation with the conditional probability is more sensible for a fixed model (see section 2.2).

- **Prospective or retrospective?** Feature importance could refer to the expected importance prior to inspecting its value, or the conditional importance after the feature was observed. This can also be expressed as global and local feature importance. With a few exceptions [20, 47], most approaches assume the local feature importance is of interest.

- **Interactions** Standard Shapley values and its variants can obscure interactions between different players in cooperative game theory or between features in machine learning. Extensions of Shapley values, like the power series formulation we developed, allow these interactions to be explicitly identified [9, 10]. As these interaction terms are defined in slightly different ways they are not immediately comparable.

- **Interpretation** Molnar [21] interprets the Shapley value as 'Given the current set of feature values, the contribution of a feature value to the difference between the actual and the mean predication is the estimated Shapley value.' This is different to how many

people intuitively interpret feature importance - as a measure of how the model performance would change in response to a feature value being censored or locally perturbed. The interpretation is also difficult to use when there is strong interactions or correlations between features, as we explored in section 4.7. Our formulation can be interpreted as the weighted change in the expected local or global loss when a feature is removed from the training or test data sets, when interactions and correlations are explicitly accounted for.

6. CONCLUSION

Statistical models are rapidly evolving in scope and sophistication. Their integration into complex systems creates new capabilities but also introduces additional risks. There is potential for significant financial damage or even loss of life if these system behave in unexpected ways. Conventional numerical measures of model performance are insufficient for guaranteeing model generalization; interpretable machine learning can provide a complementary validation by providing insight into how models determine their outputs.

We have introduced a power series representation of feature importance which explicitly quantifies the value of information. It separates individual and multi-feature contributions, and highlights synergies and redundancies in feature information. A weighted average of these contributions can be used to transform the power series into Shapley values. Our method inherits the desirable properties of Shapley values, while providing greater insight into how features interact together. Our method can identify joint contributions, such as in the exclusive-OR problem, that are ambiguous from the generic Shapley values. Our approach motivates alternative approaches to calculating the Shapley values that can speed up the calculations considerably. In particular, we showed a substantial reduction in the number of model evaluations required when only first and second-order feature interactions are present.

There are potentially wider applications of the power series formulation. Perhaps the coefficients could be used to select better subsets of features for sparse but accurate models. There may also be scope to use redundant features to construct anomaly detectors or develop error-correction processes for suppressing noise. For example, we could flag an anomaly when two features with generally high redundancy provide disparate predictions for a particular instance. The generality of the method means it is compatible with most statistical models (it does not require a specific model architecture, like a neural network, to be used), and only its basic utility has been explored in this technical report.

The literature contains several methods for defining feature importance. These can provide inconsistent views of which features are important and there is a lack of guidance in the literature about which method to choose. In our view, the power series formulation complements, rather than replaces, these alternative formulations. For example, visualization of key data points can provide insight into the data structure that may not be adequately captured by quantitative metrics. Similarly, coordinate transformations can make the features more interpretable, and can even be a pre-processing step prior to calculating feature importance. Counterfactuals are often better placed than the series expansion for suggesting possible actions. For example, it may be useful to know what features to modify so that a home loan is approved, rather than how missing data affects the quality of loan decisions. Existing methods are generally inappropriate for identifying

causal structure without additional domain knowledge, which limits their application to detecting and correcting algorithmic bias or understanding mechanisms that link features together. Domain knowledge and causal models can support the interpretation of feature importance.

Any formulation of feature importance should account for the context and the topics of interest. We highlighted this with a hypothetical example of prospectively deciding whether an x-ray is likely to yield useful information against the retrospective of how access to the x-ray did affect the medical diagnosis. Our framework can naturally accommodate a range of topics and contexts while retaining common operating principles. The greater coherence in the approach allows it to be reused more broadly than some alternative methods, and reduces the risk of generating inconsistent conclusions, which is a possibility when different principles are haphazardly employed. The improved ability to interpret data and models will assist Defence in creating and maintaining sophisticated statistical models.

7. ACKNOWLEDGEMENTS

We would like to thank Maria Athanassenas and Glennn Moy for valuable feedback and discussions.

8. REFERENCES

1. Keevers, T. L. (2019) 'Cross-validation is insufficient for model validation'. In: *https://www.dst.defence.gov.au/sites/default/files/publications/documents/DST-Group-TR-3576.pdf* **TR-3576**.
2. Doshi-Velez, F. and Kim, B. (2017) 'Towards a rigorous science of interpretable machine learning'. In: *arXiv preprint arXiv:1702.08608*.
3. Rudin, C. (2019) 'Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead'. In: *Nature Machine Intelligence* **1** (5), 206–215.
4. Breiman, L. et al. (2001) 'Statistical modeling: The two cultures (with comments and a rejoinder by the author)'. In: *Statistical science* **16** (3), 199–231.
5. Biecek, P. (2018) 'DALEX: explainers for complex predictive models in R'. In: *The Journal of Machine Learning Research* **19** (1), 3245–3249.
6. Sokol, K. and Flach, P. A. (2018) 'Glass-Box: Explaining AI Decisions With Counterfactual Statements Through Conversation With a Voice-enabled Virtual Assistant.' In: *IJCAI*, 5868–5870.
7. Robnik-Šikonja, M. and Kononenko, I. (2008) 'Explaining classifications for individual instances'. In: *IEEE Transactions on Knowledge and Data Engineering* **20** (5), 589–600.
8. Friedman, J., Hastie, T. and Tibshirani, R. (2001) *The elements of statistical learning*. Vol. 1. 10. Springer series in statistics New York.
9. Fujimoto, K., Kojadinovic, I. and Marichal, J.-L. (2006) 'Axiomatic characterizations of probabilistic and cardinal-probabilistic interaction indices'. In: *Games and Economic Behavior* **55** (1), 72–99.
10. Lundberg, S. M., Erion, G. G. and Lee, S.-I. (2018) 'Consistent individualized feature attribution for tree ensembles'. In: *arXiv preprint arXiv:1802.03888*.
11. Roubens, M (1996) 'Interaction between criteria and definition of weights in MCDA problems'. In: *44th Meeting of the European Working Group "Multicriteria Aid for Decisions", Brussels, Belgium*.
12. Grabisch, M. (1997) 'K-order additive discrete fuzzy measures and their representation'. In: *Fuzzy sets and systems* **92** (2), 167–189.
13. Harsanyi, J. C. (1963) 'A simplified bargaining model for the n-person cooperative game'. In: *International Economic Review* **4** (2), 194–220.
14. Ribeiro, M. T., Singh, S. and Guestrin, C. (2016) '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, 1135–1144.

15. Shapley, L. S. (1953) 'A value for n-person games'. In: *Contributions to the Theory of Games* **2** (28), 307–317.

16. Young, H. P. (1985) 'Monotonic solutions of cooperative games'. In: *International Journal of Game Theory* **14** (2), 65–72.

17. Lundberg, S. M. and Lee, S.-I. (2017) 'A unified approach to interpreting model predictions'. In: *Advances in Neural Information Processing Systems*, 4765–4774.

18. Sundararajan, M. and Najmi, A. (2019) 'The many Shapley values for model explanation'. In: *arXiv preprint arXiv:1908.08474*.

19. Štrumbelj, E. and Kononenko, I. (2011) 'A general method for visualizing and explaining black-box regression models'. In: *International Conference on Adaptive and Natural Computing Algorithms*. Springer, 21–30.

20. Frye, C., Feige, I. and Rowat, C. (2019) 'Asymmetric Shapley values: incorporating causal knowledge into model-agnostic explainability'. In: *arXiv preprint arXiv:1910.06358*.

21. Molnar, C. (2019) *Interpretable Machine Learning. A Guide for Making Black Box Models Explainable*. https://christophm.github.io/interpretable-ml-book/.

22. Kononenko, I. et al. (2010) 'An efficient explanation of individual classifications using game theory'. In: *Journal of Machine Learning Research* **11** (Jan), 1–18.

23. Law, A. M., Kelton, W. D. and Kelton, W. D. (2000) *Simulation modeling and analysis*. Vol. 3. McGraw-Hill New York.

24. Ancona, M., Öztireli, C. and Gross, M. (2019) 'Explaining deep neural networks with a polynomial time algorithm for shapley values approximation'. In: *arXiv preprint arXiv:1903.10992*.

25. Castro, J., Gómez, D. and Tejada, J. (2009) 'Polynomial calculation of the Shapley value based on sampling'. In: *Computers & Operations Research* **36** (5), 1726–1730.

26. Suri, N. R. and Narahari, Y. (2008) 'Determining the top-k nodes in social networks using the shapley value'. In: *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 3*. International Foundation for Autonomous Agents and Multiagent Systems, 1509–1512.

27. Hu, X. (2006) 'An asymmetric Shapley–Shubik power index'. In: *International Journal of Game Theory* **34** (2), 229–240.

28. Friedman, J. H. (2001) 'Greedy function approximation: a gradient boosting machine'. In: *Annals of statistics*, 1189–1232.

29. Goldstein, A. et al. (2015) 'Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation'. In: *Journal of Computational and Graphical Statistics* **24** (1), 44–65.

30. Apley, D. W. (2016) 'Visualizing the effects of predictor variables in black box supervised learning models'. In: *arXiv preprint arXiv:1612.08468*.

31. Higgins, J. J. (2004) *An introduction to modern nonparametric statistics*. Brooks/Cole Cengage Learning.

32. Breiman, L. (2001) 'Random forests'. In: *Machine learning* **45** (1), 5–32.

33. Wachter, S., Mittelstadt, B. and Russell, C. (2017) 'Counterfactual Explanations without Opening the Black Box: Automated Decisions and the GPDR'. In: *Harv. JL & Tech.* **31**, 841.

34. Ribeiro, M. T., Singh, S. and Guestrin, C. (2018) 'Anchors: High-precision model-agnostic explanations'. In: *Thirty-Second AAAI Conference on Artificial Intelligence*.

35. Keevers, T. L. 'Expanded Basis Sets for the Manipulation of Random Forests'. In: *Defence Operations Research Symposium (accepted)*.

36. Ba, J. and Caruana, R. (2014) 'Do deep nets really need to be deep?' In: *Advances in neural information processing systems*, 2654–2662.

37. Crowley, E. J., Gray, G. and Storkey, A. J. (2018) 'Moonshine: Distilling with cheap convolutions'. In: *Advances in Neural Information Processing Systems*, 2888–2898.

38. Papernot, N. et al. (2016) 'Distillation as a defense to adversarial perturbations against deep neural networks'. In: *2016 IEEE Symposium on Security and Privacy (SP)*. IEEE, 582–597.

39. Rudin, C., Wang, C. and Coker, B. (2018) 'The age of secrecy and unfairness in recidivism prediction'. In: *arXiv preprint arXiv:1811.00731*.

40. Adler, P. et al. (2018) 'Auditing black-box models for indirect influence'. In: *Knowledge and Information Systems* **54** (1), 95–122.

41. Bau, D. et al. (2017) 'Network dissection: Quantifying interpretability of deep visual representations'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 6541–6549.

42. Efron, B. (1982) *The jackknife, the bootstrap, and other resampling plans*. Vol. 38. Siam.

43. Freytag, A., Rodner, E. and Denzler, J. (2014) 'Selecting influential examples: Active learning with expected model output changes'. In: *European Conference on Computer Vision*. Springer, 562–577.

44. Cook, R. D. (1977) 'Detection of influential observation in linear regression'. In: *Technometrics* **19** (1), 15–18.

45. Koh, P. W. and Liang, P. (2017) 'Understanding black-box predictions via influence functions'. In: *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 1885–1894.

46. Kim, B., Khanna, R. and Koyejo, O. O. (2016) 'Examples are not enough, learn to criticize! criticism for interpretability'. In: *Advances in Neural Information Processing Systems*, 2280–2288.

47. Covert, I., Lundberg, S. and Lee, S.-I. (2020) 'Understanding Global Feature Contributions Through Additive Importance Measures'. In: *arXiv preprint arXiv:2004.00668*.

48. Grömping, U. (2007) 'Estimators of relative importance in linear regression based on variance decomposition'. In: *The American Statistician* **61** (2), 139–147.

49. Bach, S. et al. (2015) 'On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation'. In: *PloS one* **10** (7).

50. Aas, K., Jullum, M. and Løland, A. (2019) 'Explaining individual predictions when features are dependent: More accurate approximations to Shapley values'. In: *arXiv preprint arXiv:1903.10464*.

51. Štrumbelj, E. and Kononenko, I. (2014) 'Explaining prediction models and individual predictions with feature contributions'. In: *Knowledge and information systems* **41** (3), 647–665.

DEFENCE SCIENCE AND TECHNOLOGY GROUP DOCUMENT CONTROL DATA		DLM/CAVEAT (OF DOCUMENT) Unclassified
TITLE A Power Series Expansion of Feature Importance	SECURITY CLASSIFICATION (FOR UNCLASSIFIED LIMITED RELEASE USE (U/L) NEXT TO DOCUMENT CLASSIFICATION) Document (U) Title (U)	
AUTHOR(S) T. L. Keevers	PRODUCED BY Defence Science and Technology Group DST Headquarters Department of Defence F2-2-03 PO Box 7931 Canberra BC ACT 2610	
DST GROUP NUMBER DST-Group-TR-3743	TYPE OF REPORT Technical Report	DOCUMENT DATE July, 2020
TASK NUMBER	TASK SPONSOR	RESEARCH DIVISION Joint and Operations Analysis Division
MAJOR SCIENCE AND TECHNOLOGY CAPABILITY Maritime Capability Analysis	SCIENCE AND TECHNOLOGY CAPABILITY Maritime Mathematical Sciences	
SECONDARY RELEASE STATEMENT OF THIS DOCUMENT Approved for public release.		
ANNOUNCABLE No limitations		
CITABLE IN OTHER DOCUMENTS Yes		
RESEARCH LIBRARY THESAURUS Machine learning, Interpretability, Statistics		