Model Class Reliance for Random Forests

Gavin Smith
N/LAB
University of Nottingham
Nottingham, UK

Roberto Mansilla
N/LAB
University of Nottingham
Nottingham, UK
{first.last}@nottingham.ac.uk

James Goulding
N/LAB
University of Nottingham
Nottingham, UK

Abstract

Variable Importance (VI) has traditionally been cast as the process of estimating each variable's contribution to a predictive model's overall performance. Analysis of a single model instance, however, guarantees no insight into a variables relevance to underlying generative processes. Recent research has sought to address this concern via analysis of Rashomon sets - sets of alternative model instances that exhibit equivalent predictive performance to some reference model, but which take different functional forms. Measures such as Model Class Reliance (MCR) have been proposed, that are computed against Rashomon sets, in order to ascertain how much a variable must be relied on to make robust predictions, or whether alternatives exist. If MCR range is tight, we have no choice but to use a variable; if range is high then there exists competing, perhaps fairer models, that provide alternative explanations of the phenomena being examined. Applications are wide, from enabling construction of 'fairer' models in areas such as recidivism to health analytics and ethical marketing. Tractable estimation of MCR for non-linear models is currently restricted to Kernel Regression under squared loss [7]. In this paper we introduce a new technique that extends computation of Model Class Reliance (MCR) to Random Forest classifiers and regressors. The proposed approach addresses a number of open research questions, and in contrast to prior Kernel SVM MCR estimation, runs in linearithmic rather than polynomial time. Taking a fundamentally different approach to previous work, we provide a solution for this important model class, identifying situations where irrelevant covariates do not improve predictions.

1 Introduction

Post-hoc model explanation, through the use of variable importance measures, underpins a range of applications including variable selection, fit-for-purpose model auditing in domains such as criminal recidivism cases and the identification of potentially causal features. However, given input features typically share predictive information regarding the output variable, the learnt model potentially encodes just one of many equally well performing functional relationships between the input and output variables, with the choice often made arbitrarily with respect to the use case by the model building algorithm. This means variables may be incorrectly considered unimportant, that model audits are not robust to model retraining and the interpretation of potential causal features may be incomplete and misleading. In some cases these issues do not matter (i.e. minimal subset selection when predictive performance is the only goal) but often they do, with any machine learning framework/pipeline that subsequently includes the interpretation/explanation of a single model (e.g. [21, 6, 8, 5, 10]) likely to be improved.

Recently, [7] formally defined this issue - the multiplicity of models with the same prediction performance based on different underpining predicting factors - with regard to model classes. Given

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

a model class (model type and fixed complexity), the authors define a ϵ-Rashomon set as the set of all equally performing (within a tolerance of ϵ) models from a given reference model. Based on the definition of such sets the authors define two *Model Class Reliance* measures, MCR+ and MCR-. In contrast to global variable importance measures which aim to explain the degree to which a given model relies on (utilizes) each variable in the given model, MCR represents the most (MCR+) and least (MCR-) a variable is relied on in any model in the ϵ-Rashomon set. The tractable provision of upper and lower bounds of variable reliance across an entire model class, as opposed to a single model, provides information able to be utilized within the model building phase and/or in a post-hoc fashion to better evaluation the optimally of the model and the existence of alternatives.

For instance in motivating MCR, [7] utilized MCR- and MCR+ to audit models/problem domains to examine the potential reliance of models on variables considered unacceptable, a problem particularly relevant in criminal recidivism prediction. A second application is with regard to all relevant variable selection [5]. While there are well known approaches based on heuristics for these problems [11], MCR+ has a notable use-case here in providing a simple and theoretically optimal solution since if the MCR+ is zero the variable cannot be relevant. A third application is as part of the model building pipeline, enabling the incorporation of domain knowledge to either realize better models or, if exploratory analysis is being conducted provide a better understanding of the phenomenon being modelled. For instance, model reliability may be improved by selecting models that highly utilize variables known or hypothesised to be causal over others. Alternatively practitioners may consider one model to be more actionable than another, for example where predictive models are used for targeted communications with the marketing domain. Here the model itself being used to select who to target and the explanation is used to inform communication strategy and/or design. Being aware of equally predictive models enables a more relevant choice based on knowledge of the specific problem not available to the model building algorithm.

While MCR and the underpinning concept of Rashomon sets offer significant potential, the initial work by [7] provides only a limited number of methods, for a restricted number of model classes. These include models that have convex loss functions in the model parameters and linear and Kernel SVM regression with squared loss functions. Notably the former is considered potentially intractable by the authors in the general case and the latter (for which they provide a RBF Kernel implementation with polynomial runtime complexity) is the only general purpose non-linear model class. Acknowledging the limitations directly, the authors leave as open problems more general purpose non-linear algorithms, particularly noting model classes in which irrelevant covariates do not improve predictions hypothesising these many be potentially intractable [7, pg. 19, Table 1].

In this work we directly address this gap, providing a method to compute MCR for both classification and regression Random Forests with linearithmic runtime complexity. The use of Random Forests additionally provides a higher utility in practice as they are comparatively robust to correlated and/or irrelevant variables, thereby often being a better choice of model class for the reference model. This is particularly true given (1) alternate explanations in part stem from correlated features and (2) the distinction between irrelevant vs. relevant variables is unknown when attempting to learn (alternate) explanations.

2 Comparison of MCR with existing VI methods

Model Class Reliance aims to measure fundamentally different (but related) information to existing variable importance techniques. Methods such as unconditional permutation importance [3] and recent variants [7] as well as methods such as Gain [2], Split Counts [4] and Minimal Depth [9] which leverage the structure of tree based models seek only to measure the importance of a variable with regard to a single predictive model. Recently Minimal Depth was extended by [18] such that a variable's importance, measured by its depth, also included when it could have been used to realize a split identically to the chosen split. This can be seen as enabling a single tree to represent many others from the same class (and our approach leverages this insight in part). As will be discussed, however, this is not enough to achieve tight MCR bounds. Moreover, the approach by [18], being a measure based on tree structures is known to be inconsistent [15] reporting incomparable attribution scores as models change their structure making the results difficult to interpret in many use cases.

SHAP values [14, 15] by design measure a variable's average contribution and therefore are not suitable for measuring contribution bounds. Specifically, seeking the maximum or minimum contribution for a variable breaks the first property, local accuracy, from which SHAP is derived. We note that this is not deficit of SHAP, SHAP fulfils its aim to explain a specific model instance, but a fact that highlights that SHAP is unable to be re-purposed as an MCR measure. Algorithm Reliance (AR) [7] methods retrain models to determine the effect of holding out variables. [7] consider two types of AR, *AR necessity* and *AR sufficiency*. The former considers hold out of the variable of interest, and the latter the holding out of all variables except the one of interest. While *AR necessity* has conceptual similarities to MCR-, AR methods provide a different type of measurement, considering a larger class of models, and including candidates that do not fit the data equally well [7][1]. Moreover, the computational requirement to retrain a model per variable of interest can render them impractical.

Conditional variable importance [20] is a related permutation importance measure which disrupts a variable conditional on full information of all other variables. If information is shared between the variable of interest and other variables this reduces the impact of the permutation to the appropriate degree. Therefore, as long as the conditional distribution can be correctly approximated conditional permutation importance provides a similar measure to MCR-. The fact that it is based on a single model is not important here as the fact that a variable could have been used in a different model in the class is irrelevant, and it is desirable that no effect is recorded for this variable. In practice, however, approximating the conditional distribution can be error prone in the case of real valued datasets.

The most related work to MCR+ is [1, 16] which seek to identify *indirect influence* for black-model auditing. In contrast to MCR+ these methods are designed to focus solely on auditing a single model by, for a given variable of interest, 'noising up' the other input variables to remove any shared information and then measuring the loss in predictive performance. The proposed algorithms learn to disrupt relationships between the variable of interest and input variables independently and current approaches are either overly simplistic (considering only pairwise variable relationships [1]) or required significant computation time and the tuning of complex of meta-parameters [16] resulting in different levels of error in the learnt relationship per variable. Without significant effort in tuning the algorithms, the potential for differing levels of error confounding the reported variable's importance make their interpretation somewhat tenuous. In summary, while related, the methods are both not designed to address the same problem as MCR+ and are currently complex to use in practice.

3 Model Class Reliance Preliminaries

In this section we introduce the notation required for us to define MCR estimators for Random Forest model classes. In general, we will broadly accord to that used in [7], in referring to a predictive model, $F \in \mathcal{F}$, that has been constructed using a dataset generated from a random variable $Z = (X_1, X_2, Y) \in \mathcal{Z}$, where our target variable is $Y \in \mathcal{Y}$, and our predictors are sets of covariates X_1 and X_2. These input variables are separated merely to allow us to focus on X_1 in notation as the (potentially multivariate) variable whose importance is being investigated, with X_2 being the remaining input features. To derive our estimators, we first define several key terms:

Model Reliance: reflects the extent to which a single fitted model relies on variable X_1 to achieve its predictive performance, and is denoted as $MR_{X_1}(F)$. To calculate model reliance, we first apply a user-defined non-negative loss function $L : \mathcal{F} * \mathcal{Z} \to \mathbb{R}_{\geq 0}$ to assess the model's predictive performance. Next we disrupt X_1 using some specified function, $\Phi : \mathcal{Z} \to \mathcal{Z}$, in order to render the variable as uninformative as possible for prediction, while keeping the same marginal distribution. The new dataset after disruption, $Z^{\Phi} = (X_1^{\Phi}, X_2, Y)$, is then used to obtain a new expected loss score, $\mathbb{E}L(F, Z^{\Phi})$. Model reliance is then derived as either the ratio or the difference between the two expected loss scores (before/after disruption) [7]. In this study we have used the difference:

$$MR_{X_1}(f, \Phi) = \mathbb{E}L(f, \langle X_1^{\phi}, X_2, Y \rangle) - \mathbb{E}L(f, \langle X_1, X_2, Y \rangle) \quad (1)$$

Model reliance can be interpreted as the loss introduced when we destroy X_1's predictive capacity. However, final interpretation relies on the choice of disruption function, Φ. MR was introduced in [7] using a 'switch' permutation strategy, Φ_S where all possible permutations of X_1 with $\langle X_2, Y \rangle$ that do not exist in the original dataset, Z, are exhaustively constructed to form Z^{Φ}. However, the notion of model reliance is generalizable to other disruption strategies (e.g. unconditional permutation [3]). Note also that the size of the disrupted dataset, Z^{Φ}, will be greater than or equal to the original given

[1] The difference between AR and MCR is discussed in more detail in the Supplementary Material.

that $|Z^{\Phi}| = n|Z|$, where n reflects the number of permutation rounds specified when Φ_U or Φ_C are used, or exactly $n = |Z| - 1$ for the exhaustive permutation strategy used by Φ_S.

Rashomon Set: Model reliance tells us only about the contribution X_1 makes to a single model instance. To understand model reliance across a whole class of predictors, we must examine the range of MR values which exist across all equally well performing predictors. This is the ϵ-Rashomon set, and given a reference model F, is the set, $\mathcal{R}_\epsilon(F)$, of all models with a predictive performance within ϵ of F [7]. For most practical applications F refers to the best performing model found via cross-validation (or some equivalent testing procedure).

Maximum Model Class Reliance (MCR^+): The maximum possible change in predictive performance that could be attributed to our variable of interest, X_1, in any model within the ϵ-Rashomon set, $R_\epsilon(F)$. Note that MCR^+ corresponds to the MR of a realizable model, and is defined as:

$$MCR^+_{X_1}(F, \Phi) = \max_{F' \in R_\epsilon(F)} (\ MR_{X_1}(F', \Phi)\ ) \tag{2}$$

Minimum Model Class Reliance (MCR^-): In a similar manner, we can examine the minimum possible change in predictive performance attributable to X_1 for any model in the ϵ-Rashomon set:

$$MCR^-_{X_1}(F, \Phi) = \min_{F' \in R_\epsilon(F)} (\ MR_{X_1}(F', \Phi)\ ) \tag{3}$$

Empirical MCR ($\widehat{MCR}$): The range, MCR, formed from [MCR^-,MCR^+] of course refers to population level statistics, which we cannot obtain with our sample dataset, Z. Thus estimators for all of the above must be used in practice, with MR being estimated using our in-sample loss rather than expected loss over the whole population. This is referred to as our empirical MCR:

$$\widehat{MCR}_{X_1}(F, \Phi) = \left[\min_{F' \in \hat{R}_\epsilon(F)} \left(\ \widehat{MR}_{X_1}(F', \Phi)\ \right),\ \max_{F' \in \hat{R}_\epsilon(F)} \left(\ \widehat{MR}_{X_1}(F', \Phi)\ \right) \right] \tag{4}$$

4 A novel MCR method for Random Forests

Although an $\widehat{MCR}$ range is fully determined by a sample, actually computing it in practice is highly non-trivial, mostly due to the maximization and minimization steps required in equations 2 and 3 respectively. This challenge was recognized in [7], which introduced a polynomial-time optimization procedure to estimate MCR for Kernel-based SVM classes. We now present a novel method for estimating MCR for Random Forest model classes, that functions in linearithmic time.

Recall that we seek to find the maximum class reliance, $MCR^+_{X_1}(F)$ and minimum, $MCR^-_{X_1}(F)$, across all models in an ϵ-Rashomon set, $\mathcal{R}_\epsilon(.)$, with reference to some variable of interest, X_1. The Rashomon set is itself established based on a preexisting reference model instance, F, of some fixed complexity, θ, and trained using data, $Z = \langle X_1, X_2, Y \rangle$. Here X_2 represents additional input features accompanying X_1, and Y the output feature. We reasonably assume it is impossible to iterate over every element in $\mathcal{R}_\epsilon(F)$ to obtain true MCR values directly due to computational intractability.

We therefore instead seek estimates for $\widehat{MCR}^+(F)$ and $\widehat{MCR}^-(F)$ that are both tight and obtainable within reasonable time complexity. In order to derive such estimators, we take a divide-and-conquer approach, conceptually mapping each tree in the reference model to a new tree which maximises (or minimizes) reliance on X_1 while maintaining predictive equivalence with to F (and hence membership of the Rashomon set, $R_\epsilon(F)$). Our strategy for achieving this traverses an estimate of each individual tree's own Rashomon set, $\hat{R}_\epsilon(f)$, via application of two key transformations, as follows:

4.1 Estimation of MCR^+

For each tree in the forest, $f \in F$, its individual Rashomon set, $R_\epsilon(f)$, is examined to find a new tree that is maximally reliant on X_1 yet has identical structure and makes the same predictions across all data (a characteristic henceforth referred to as *structural equivalence*). Despite the strictness of this mapping, searching the whole of $R_\epsilon(f)$ is computationally intractable, so we instead apply a transformation function, $\mathcal{S}^+(f)$ that leverages *surrogate splits* [2]. This function examines each node in f, and replaces the existing decision variable with X_1 if they are surrogates (identified during construction of F [18]). This allows us to directly identify a preferred tree in $R_\epsilon(f)$, from the $m^{\frac{n}{2} \leq n\_splits \leq n-1}$ structurally equivalent trees to f that exist [13] where m is the number of input features and $n = |Z|$ is the number of data points. We are guaranteed due to this structural equivalence that:

$$\forall f \in F : \widehat{MR}_{X_1}(\mathcal{S}^+(f)) \geq \widehat{MR}_{X_1}(f) \tag{5}$$

Once this transformation has been applied to all trees, we can construct an entirely new random forest, $F_S^+ = \{S^+(f) : f \in F\}$. By construction F^+ will produce exactly the same predictive outputs as F, despite containing components with greater usage of X_1, and hence:

$$MR_{X_1}(F_S^+) \geq MR_{X_1}(F) \tag{6}$$

Having formulated F_S^+, the next step in estimating $MCR^+(F)$ is to relax the requirement for structural equivalence altogether. To implement this we identify a random forest in $R_\epsilon(F)$, that uses trees from F_S^+ but in a new combination (and allows for duplication) that maximizes reliance on X_1 but which maintains the same prediction accuracy.

This is achieved via a second transformation $T^+(F_S^+)$, mapping each tree in F_S^+ to some other in F_S^+ that produces the same predictions but which leverages our variable of interest the most (n.b. if a tree is already the most reliant on X_1 it is mapped to itself). To formalize this, let us first define $\Psi(f)$ as the set of trees that share *predictive equivalence* with f, given dataset Z:

$$\Psi(f) = \{f' \in F : \forall_{x \in Z} \; f(x) = f'(x)\} \tag{7}$$

Note that this notion of predictive equivalence between models, Ψ, doesn't require they share any structural similarities whatsoever, nor that their predictions even be correct - but merely that they are the same for all data in Z. Given this, we can succinctly define $T^+(f)$ as:

$$T^+(f) = \underset{f' \in \Psi(f)}{\arg\max} \; MR_{X_1}(f') \tag{8}$$

Applying this transformation to map every tree to another in its set of predictive equivalents that has maximal MR, generates a co-domain that we can collate to construct our boundary model:

$$F_T^+ = \{T^+(f) : f \in F_S^+\} \tag{9}$$

This model remains in the Rashomon set of F, and importantly we remain guaranteed that $MR_{X_1}(F_T^+) \geq MR_{X_1}(F_S^+)$ (please see supplementary materials for the full proof). To obtain our our final estimator for MCR^+ we simply assess this model's reliance on X_1:

$$\widehat{MCR}^+_{X_1}(F) = MR_{X_1}(F_T^+) \tag{10}$$

With $\widehat{MCR}^+$ in hand, the question remains as to why this reflects a good estimator of the class' maximum reliance on X_1. To demonstrate this, consider first that as the number of trees in F tends to infinity, all possible constructable trees will occur in its ensemble (due to the stochastic nature of its generating algorithm). This means the transformation $T^+(f)$ will be replacing every tree that can possibly exist, with the tree from its complete Rashomon set that holds maximum reliance on X_1 for any potential composition of forest. Thus, as $|F| \to \infty$, our estimator $\widehat{MCR}^+(F)$ can only increase monotonically towards MCR^+. However, given we only ever have finite models in practice the question remains - *how quickly does it increase?* The answer to this is to some extent dependent on the size of the dataset, |Z|, and the reference tree's complexity constraints, Θ. However, we will show empirically in §5.3 that for practical purposes our method converges to a consistent asymptotic value for the estimator for only a moderate numbers of trees. This will provide evidence that a good estimator is likely to been obtained in the majority of cases. On synthetic data where the solution is known we will further show that the model is able to recover the true MCR^+ value almost precisely.

4.2 Estimation of MCR^-

An estimator of MCR^-, the minimum model class reliance on X_1, can be derived in similar fashion to that demonstrated in §4.1. However, in this case we first apply transformation $\mathcal{S}^-(f)$ to every tree in the reference model, F, a mapping that again implements surrogate splits, but which removes (rather than injecting) X_1 from any decision node when a surrogate for it is available. This guarantees:

$$\forall f \in F : \widehat{MR}_{X_1}(\mathcal{S}^{-}(f)) \leq \widehat{MR}_{X_1}(f) \tag{11}$$

Once applied to all $f \in F$, we can again conceptualize a forest that contains only these minimally reliant trees, F_S^-. We then apply a second transformation, T^- (again in symmetry with our derivation of MCR^+), that substitutes any tree in F with another that is in its set of predictive equivalent trees, Φ, but which has the least reliance on X_1:

$$T^{-}(f) = \underset{f' \in \Psi(f)}{\arg\min}\ MR_{X_1}(f') \tag{12}$$

Once this transformation is applied to all trees in F_S^-, we are left with a model, F_T^-, which remains firmly in the reference model's Rashomon Set while having minimal reliance on X_1. As before our our final estimator MCR^- can then be calculated as this boundary model's overall reliance on X_1:

$$\widehat{MCR}^{-}_{X_1}(F) = MR_{X_1}(F_T^-) \tag{13}$$

4.3 Computational of RF-MCR in practice

The computational complexity for computing RF-MCR is $\Theta(K\tilde{N}log_2(N))$ for both regression and classification. $\tilde{N}$ is the size of the set MCR is being evaluated on (training or a held out set). This assumes the reference model is built using a modified Random Forest algorithm in which surrogates are recorded (requiring all variables to be considered as potential splits at each node even if they are not being considered as a main split) and the final instance predictions per tree used to record prediction equivalent tree groupings (Ψ, Eqn 7), costing kn operations using a hashmap. The latter requires bootstrapping be avoided and randomness introduced into the forest via variable split selection. Complexity is unchanged from existing algorithms with these constraints: $O(mkn\ log_2 n)$.

Computation of RF-MCR is done in 3 steps. Step 1 computes the $MR(S^+(f))$ for each tree by computing unconditional VI but with forced surrogate use (MCR+ or forced avoidance, MCR-) with regard to the variable of interest. This involves making a prediction per data point and can be done in $K\tilde{N}log_2(N)$. The result is the set F_S^+ and associated MR scores. Step (2) leverages Ψ and $MR(S^+(f))$ to compute F_T^+ (Equation 9). This can be done in K time by iterating over the groups of trees that are predictive equivalent and, as a replacement for all trees in that group, selecting the one with the highest MR. The final step (3) involves computing $MR(F_T^+)$ which takes $K\tilde{N}log_2(N)$.

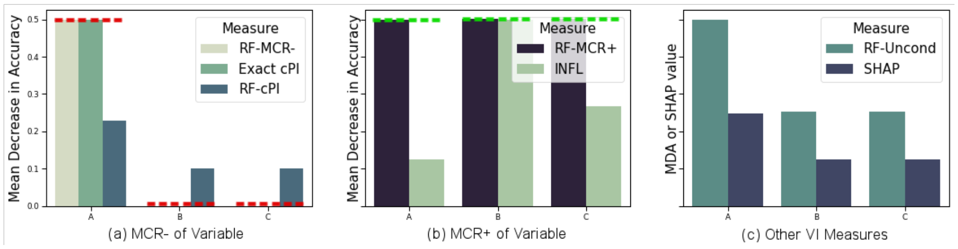
So far only the computation of 0-Rashomon sets has been considered. The approach can be extended to a ϵ-Rashomon set by: (1) relaxing the requirement for surrogate splits to be exact [18] and (2) the relaxation of exact prediction equivalence when forming Ψ. For (1) is not clear how this could be easily linked to a desired ϵ value and as such this extension is left as future work and only extension (2) implemented in this work. This does not introduce any additional runtime complexity.

Finally, as noted in Section 3, the proof for Empirical MCR is based upon in-sample loss, rather than the expected loss over the whole population. This leaves an outstanding question as to whether population level MCR+/- values can be accurately determined by constructing/searching over ϵ-Rashomon sets identified from a sample. To this end [7] provide an applicable theoretical proof, demonstrating that in-sample estimation of MCR is sufficient under large sample sizes, assuming a reference model with a correctly selected complexity [7, §4.1]. This proof corresponds to the real-world use case where MCR is to be computed based on a reference model that has been trained on the full dataset (following the frequent strategy of applying cross-validation in order to determine a model complexity, prior to refitting on all data to produce final model outputs). In particular, this approach is commonplace when computing traditional variable importance measures (i.e. via permutation) [17]. A thorough investigation into the use of MCR within a sample splitting approach [7] remains interesting future work.

5 Empirical Evaluation

In this section we demonstrate: the ability of RF-MCR[2] to compute $\widehat{MCR}$ bounds in a tractable fashion; the convergence of the estimators; the ability to recover true MCR values; and the insights derivable on 2 real world datasets (recidivism and breast cancer). A third real-world application (negative birth outcomes in East Africa) is additionally considered in the supplementary material.

First we consider data sampled from a simple generative network, to show ability to recover true MCR bounds. The network is setup such that variables A and B form an XOR predictive relationship, meaning *both* are required to predict Y above random chance (50% accuracy). If both are known, then a perfect prediction can be made. Additionally, a variable C is generated so as to be perfectly colinear with B. The true MCR range (with MR measured as mean decrease in accuracy) for A is [50%, 50%], and B and C both have true MCR of [0%,50%]. A dataset of 1000 instances is generated from this network, to which various variable importance approaches are applied alongside RF-MCR. As this data forms a non-linear *classification* problem, SVM-MCR [7] cannot be applied.



(a) MCR- of Variable

(b) MCR+ of Variable

(c) Other VI Measures

Figure 1: Synthetic Dataset Results: (a) MCR^{-} recovery (b) MCR^{+} recovery (C) VI methods. True values or MCR- and MCR+ are indicated in red and green respectively

As shown in Figure 1(a), both our approach and conditional permutation importance (Exact-cPI) generate values that correspond perfectly to the true MCR^{-}. The approximation of cPI for conditional random forests (RF-cPI) [20] is also shown, although its approximation errors are evident. Figure 1(b) assesses the identification of MCR+, comparing RF-MCR to the previously mentioned disentanglement technique in [16] (which we refer to as INFL here). Again, RF-MCR recovers the true values for each variable correctly. However, the neural networks used by INFL seem unable to correctly learn the underlying relationship despite significant tuning (full details of tuning are in the Supplementary Material), significantly underestimating the ability of A and C to contribute to successful predictions. Finally, Figure 1(c) shows that two other VI approaches, SHAP and unconditional permutation importance, offer no insight into MCR^{-} and MCR^{+} as expected.

These results provide initial evidence that RF-MCR can recover true model class reliance values - and the inability of traditional VI methods to reflect this range. However, they do indicate some correspondence between cPI and INFL with $MCR-$ and $MCR+$ respectively. In the following experiments we combine these two measures to form a HYBRID-MCR method solely for purposes of comparison. For regression, where MCR for Kernel Regression under squared loss (SVM-MCR) [7] has been previously proposed, we compare to both HYBRID-MCR and SVM-MCR.

5.1 COMPAS - Recidivism Modelling

Next we examine the effectiveness of RF-MCR on a real world regression dataset that aims to predict the sensitive issue of recidivism (probability of re-offending) using the same dataset as [7]. We replicate their results for SVM-MCR on the COMPAS data, and extend their analysis from SVMs to Random Forest classes. The COMPAS system provides records from a set of defendants from Broward County, Florida labelled with known recidivism scores [12]. The goal is to understand if unfair and potentially inadmissible variables, such as race, might be being unfairly relied on by models. Assuming a reference model of sufficient predictive performance can be obtained, identifying a high $MCR+$ in analysis tells us a variable may be being heavily leveraged in predictions; a low MCR^{-} score squarely indicates proxies for X_1 exist, and its use need not be the case.

[2] A python version of the implementation is available on github: `https://github.com/gavin-s-smith/mcrforest`.

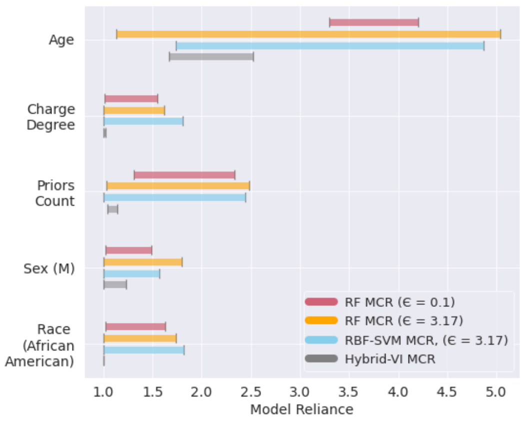


Figure 2: COMPAS dataset MCR Results

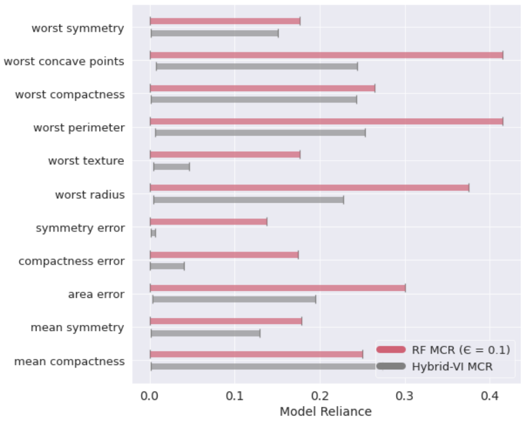


Figure 3: Breast Cancer dataset MCR results

Figure 2 shows the results. Both SVM-MCR and RF-MCR methods utilize reference models with similar mean squared errors (2.86 and 2.73 respectively) on a held out test set. In practice the SVM-MCR requires $\epsilon > 0$, so is set at $\epsilon = 3.17$ as per [7]. RF-MCR is tested at $\epsilon = 0$ (where it can be compared to the unparameterizable Hybrid-MCR), but we additionally test RF-MCR at $\epsilon = 3.17$ for direct comparison with SVM-MCR. Despite using a definition of MR in §4 as the difference between model error before and after variable disruption (which we find more intuitive), we report results using their ratio to aid direct comparison with the results in [7]. Following [7] we also left truncate results, and to aid comparison of approaches present results all input variables individually.

The ranges in Figure 2 immediately highlight the similar results for SVM-MCR and RF-MCR (at $\epsilon = 3.16$). There are differences, as SVM and RF classes afford different model solutions, but their general agreement adds support to the fact that both techniques are providing good estimators of true MCR (for more see §5.3). Most importantly, we now have 2 model classes supporting the fact that race is a non-necessary variable in recidivism modelling. However, RF-MCR adds increased insight that age can also shares mutual information with other variables that might be used instead. The difference between age MCR for $\epsilon = 3.17$ and $\epsilon = 0$ is stark, but illustrates that as the Rashomon set shrinks, so does MCR range - with some variables being far more sensitive to this effect. Our Hybrid-MCR baseline illustrates the limits of using VI measures. The fact its lower bound is similar to SVM-MCR is interesting (and requires further analysis), but it clear that the upper bound, INFL [16] is a poor estimate - not unexpected given it was not built for this purpose, but emphasizing the value of MCR to obtain domain insights into an underlying domain.

5.2 Breast Cancer Dataset

The Breast Cancer Wisconsin dataset [19] is a dataset from the UCI ML Repository, well known as having input variables with shared information. The dataset reflects a classification task to predict a tumour as malignant or benign. While well examined, this is the first time model class reliance could be applied to the dataset. Results in Figure 3 highlight the wide MCR ranges that exist across almost all variables, emphasizing the ability of RF-MCR to rapidly identify non-linear correlations. No variable is uniquely predictive within it; and only a few variables (with small ranges) offer little contribution to the predictive task at all. Given the good predictive performance of resulting models (95.74%), this means that multiple, distinct combinations of variables can produce equally performant models. For each feature, there exists some model that achieves maximal predictive performance where that feature is not required. While MCR currently provides overarching insights of this nature, determining the minimal subsets of variables for which equally performant models exists remains interesting future work. MCR analysis of this data does, however, demonstrate that when variable importance analysis is applied to a single model misleading results can occur - especially if one is looking for insight into some underlying generative process. In such cases, one of many predictive relationships can be overemphasized - hiding a variable of potential importance form practitioners, and/or conveying to them an incorrect understanding of the problem domain itself.

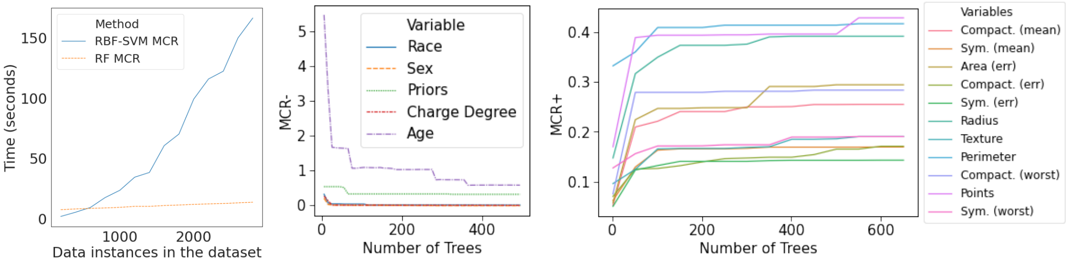
Comparing the RF-MCR to other baselines we note that no other non-linear MCR method exists for classification, so we can only contrast to our Hybrid-MCR baseline, which severely underestimates in its attempt to bound MCR+ (but please see supplementary material for error graphs, which indicate a relatively good disentanglement model was obtained, despite significant per feature variation in the noise and non-trivial reconstruction errors).

5.3 RF-MCR Analysis

Demonstrating the applicability of RF MCR we provide an empirical evaluation of runtime on the COMPAS dataset comparing to the polynomial time RBF-SVM MCR. The results are summarized in Figure 4(a) (see Supplementary material for more detail of results and setup) and report the time to compute the MCR range for an individual variable in seconds as the size of the dataset (train/test of equal size) is varied. The results clearly show the linearithmic nature of the implementation.

In Section 4.1 it was theoretically motivated that RF-MCR will quickly increase towards MCR+ as the number of trees tends to infinity (and vice-versa for MCR-). The question remains as to how quickly this occurs in practice. Figure 4 (b) and (c) provide empirical evidence for this for MCR- (COMPAS dataset) and MCR+ (Cancer dataset). The full set of MCR-/+ graphs are included in the Supplementary Material and reflect similar results - that as expected quick convergence to a point of stability for all variables is obtained with a moderate number of trees.

Finally an investigation was performed into the effect of the two transforms: the use of surrogate splits and prediction equivalence of trees by comparing the MCR results to an ablated version where prediction equivalence trees were not considered. Traditional permutation importance on the reference model used as the baseline. The results showed that the use of surrogates identified between $1.11 - 15.38\%$ of the increase in the reported upper bound importance (MCR+) with prediction tree equivalence responsible for the rest. For MCR- the decrease attributed to the use of surrogates was between $2.06 - 100\%$ with prediction tree equivalence responsible for the rest. Full details and results are included in the supplementary material. The results empirically demonstrate the requirement for both components of the algorithm.



(a) Runtime Performance (b) MCR- (COMPAS) (c) MCR+ (Cancer)

Figure 4: RF-MCR Analysis Results

6 Conclusion

Model Class Reliance (MCR) provides a more comprehensive measure of variable importance than traditional methods. Considering all models within a model class, rather that the one model the model building algorithm realised, reveals important information about possible predictive relationships between a model's input and output features that practitioners may want to leverage or avoid.

In this work we significantly extend the utility and applicability of MCR in practice, introducing novel algorithms to estimate MCR for the Random Forest classification and regression model classes for the first time. In contrast to the only other known method for computing MCR in the non-linear case (Kernel SVM Regression), the proposed approach (1) includes non-linear classification for the first time and (2) significantly reduces run-time complexity from polynomial to logarithmic time enabling its use on large datasets. Evaluations on one synthetic and two real world datasets demonstrates the utility of the approach in practice.

Broader Impact

The proposed work has the potential for significant societal impact. Understanding alternative predictive relationships that machine learnt models could equally be employing in their decision making while still achieving the same predictive performance is important for many applications. One particular case is when machine learnt models are then used in a way that significantly affects people's lives, such as the recidivism example detailed in this work. As such the approach has a part to play in ensuring algorithm fairness and reducing bias, including in model and data auditing. Other important use cases include scientific exploration of potentially causal factors in large datasets which then can form hypotheses for future causal studies. Without such methods proposed here that can be executed tractably and easily by those outside the field of computer science, the use of machine learning models may lead to misunderstanding if a practitioner assumes the relationship identified by a model is the only one or concludes that their hypothesised relationship does not exist or is not well performing when it is. A similar argument can be made with regard for the need for such techniques in business applications. Again as is discussed in the paper, the approach has applications for marketing and communication. The work could of course be used negatively. Rather than identifying and selecting the model from the set of best models in a principled way that benefits humanity, people could use it to select a model that furthers their aims at the detriment of others while still being able to claim they selected a performance optimal model. When releasing the model in a proprietary package this may go unnoticed.

Acknowledgments and Disclosure of Funding

This work was supported by the following grants: Risk prediction for Women's Health and Rights in Tanzania: novel statistical methodology to target effective interventions (EP/T003928/1) and From Human Data to Personal Experience (EP/M02315X/1).

References

[1] Philip Adler, Casey Falk, Sorelle A Friedler, Tionney Nix, Gabriel Rybeck, Carlos Scheidegger, Brandon Smith, and Suresh Venkatasubramanian. Auditing black-box models for indirect influence. *Knowledge and Information Systems*, 54(1):95–122, 2018.

[2] L. Breiman. *Classification and regression trees*. Wadsworth statistics/probability series. Wadsworth International Group, 1984. ISBN 9780534980535. URL `https://books.google.co.uk/books?id=uxPvAAAAMAAJ`.

[3] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[4] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794, 2016.

[5] Frauke Degenhardt, Stephan Seifert, and Silke Szymczak. Evaluation of variable selection methods for random forests and omics data sets. *Briefings in bioinformatics*, 20(2):492–503, 2019.

[6] Gregor Engelmann, Gavin Smith, and James Goulding. The unbanked and poverty: Predicting area-level socio-economic vulnerability from m-money transactions. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 1357–1366. IEEE, 2018.

[7] Aaron Fisher, Cynthia Rudin, and Francesca Dominici. All models are wrong, but many are useful: Learning a variable's importance by studying an entire class of prediction models simultaneously. *Journal of Machine Learning Research*, 20(177):1–81, 2019.

[8] Deanna Greenstein, Brian Weisinger, James D Malley, Liv Clasen, and Nitin Gogtay. Using multivariate machine learning methods and structural mri to classify childhood onset schizophrenia and healthy controls. *Frontiers in psychiatry*, 3:53, 2012.

[9] Hemant Ishwaran, Udaya B Kogalur, Eiran Z Gorodeski, Andy J Minn, and Michael S Lauer. High-dimensional variable selection for survival data. *Journal of the American Statistical Association*, 105(489):205–217, 2010.

[10] Hemant Ishwaran et al. Variable importance in binary regression trees and forests. *Electronic Journal of Statistics*, 1:519–537, 2007.

[11] Miron B Kursa, Witold R Rudnicki, et al. Feature selection with the boruta package. *J Stat Softw*, 36(11):1–13, 2010.

[12] Jeff Larson, Surya Mattu, Lauren Kichner, and Julia Angwin. How we analyzed the COMPAS recidivism algorithm, 2016. URL `https://www.propublica.org/article/how-we-analyzed-the-compas-recidivism-algorithm`.

[13] Gilles Louppe. *Understanding Random Forests: From Theory to Practice*. PhD thesis, Université de Liège, Liège, Belgique, 2014.

[14] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*, pages 4765–4774, 2017.

[15] Scott M Lundberg, Gabriel G Erion, and Su-In Lee. Consistent individualized feature attribution for tree ensembles. *arXiv preprint arXiv:1802.03888*, 2018.

[16] Charles Marx, Richard Phillips, Sorelle Friedler, Carlos Scheidegger, and Suresh Venkatasubramanian. Disentangling influence: Using disentangled representations to audit model predictions. In *Advances in Neural Information Processing Systems*, pages 4498–4508, 2019.

[17] Christoph Molnar. *Interpretable machine learning*. Lulu. com, 2019.

[18] Stephan Seifert, Sven Gundlach, and Silke Szymczak. Surrogate minimal depth as an importance measure for variables in random forests. *Bioinformatics*, 35(19):3663–3671, 2019.

[19] W Nick Street, William H Wolberg, and Olvi L Mangasarian. Nuclear feature extraction for breast tumor diagnosis. In *Biomedical image processing and biomedical visualization*, volume 1905, pages 861–870. International Society for Optics and Photonics, 1993.

[20] Carolin Strobl, Anne-Laure Boulesteix, Thomas Kneib, Thomas Augustin, and Achim Zeileis. Conditional variable importance for random forests. *BMC bioinformatics*, 9(1):307, 2008.

[21] John Joseph Valletta, Colin Torney, Michael Kings, Alex Thornton, and Joah Madden. Applications of machine learning in animal behaviour studies. *Animal Behaviour*, 124:203–220, 2017.

Supplementary Material: Model Class Reliance for Random Forests

Gavin Smith
N/LAB
University of Nottingham
Nottingham, UK

Roberto Mansilla
N/LAB
University of Nottingham
Nottingham, UK
`{first.last}@nottingham.ac.uk`

James Goulding
N/LAB
University of Nottingham
Nottingham, UK

1 Implementation Availability

Our RF-MCR method for both Classification and Regression Random Forests is available as a pip installable python3 package on github: `https://github.com/gavin-s-smith/mcrforest`

2 Empirical Evaluation

All experiments were run from a Google Colaboratory notebook connected to a local instance utilizing an Intel® Core$^{\text{TM}}$ i7-3930K CPU @ 3.20GHz with 64GB RAM and running Ubuntu 20.04, Python 3.6.10 and R 3.6.3. Unless otherwise specified all algorithms were timed on single core versions even though, for instance, the proposed method is in places trivially parallelizable (i.e. during forest build). An exception was the grid search across meta-parameters to find the best (optimal) reference model where parallelization was used when required as this stage does not form part of the time comparisons.

2.1 Notes for Replication of Results

Replication is facilitated through the provision of four hosted Python notebooks which replicate the paper results. Hosted on Google Colaboratory they enable the use of hosted or local runtime environments. When tested hosted runtimes were running Python 3.6.9 and R 3.6.3. Please note that while a hosted runtime can be used for ease of replication, all timings reported in the paper were based on using a local runtime environment as previously indicated NOT a hosted environment.

The URLs for the notebooks are:

1. Synthetic Experiments:
   `https://colab.research.google.com/drive/1nHaP8iNnTyY8txptNASWT2fjRPvrOSy4`
2. COMPAS Experiments:
   `https://colab.research.google.com/drive/1JiJpC8KJAeALt1wxmluUWCCGnkWfsZCG`
3. Breast Cancer Experiments:
   `https://colab.research.google.com/drive/1VYzSf9vg6-kzQHJwdi-vUrwvjr-djqjQ`
4. RF-MCR Analysis:
   `https://colab.research.google.com/drive/1BFzoR9SDNRcKv1R9BX7dnRMcIG83TYdX`

The notebooks, when run in the hosted environment will automatically install the required packages developed as part of this work. The packages developed as part of this work are discussed below and made available via the above notebooks. Note that the learning of meta-parameters for the Disentangling Influence method from [4] is generally to time consuming for a hosted notebook and the parameters from a prior run have been included. By default the notebooks will use these and skip the meta-parameter search.

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

2.1.1 The proposed method RF-MCR source code

The code is written as an extension to the sklearn RandomForestRegressor and RandomForestClassifer classes. To simplify installation, the implementation has been decoupled from sklearn and packaged as a separate installable library. The code is available on github: `https://github.com/gavin-s-smith/mcrforest`

Dependencies, download links and install instructions are included in the notebooks in the case a local install is desired. If running the notebooks on a hosted instance this will be automatically installed.

2.1.2 A Python wrapper for the non-linear RBF-SVM MCR method from [1]

The wrapper calls the R code from the lead author's github `https://github.com/aaronjfisher/mcr` and derived from `https://github.com/aaronjfisher/mcr-supplement`.

Dependencies, download links and install instructions are included in the notebooks in the case a local install is desired. If running the notebooks on a hosted instance this will be automatically installed.

2.1.3 A Python wrapper for the Disentangling Influence method from [4]

The wrapper is based on the authors code from `https://github.com/charliemarx/disentangling-influence`. The wrapper extends the code base from the author to:

- include code to grid search for the neural network meta-parameters to find the optimal parameters per (encoder, discriminator, decoder) triple, noting that there is one triple per feature of interest. This code is parallelized. For the encoder and decoder neural networks two layers were used (reflecting [4]). The grid search considered 80 different model meta-parameterizations:
  - four values for the latent dimension equally spaced between the number of input features and the cardinally of output.
  - four values for each of the hidden layers in the encoder and decoder selected to be equally spaced between the latent dimension and two times the number of covariates. For the encoder, combinations of these for the two layers such that the first layer had more neurons than the second were considered. For the decoder reversed combinations were considered.
  - two different learning rates were considered 0.001 and 0.01
  - beta was set at 0.5 or 1 based on discussion in [4] and preliminary experimentation
  - the maximum number of training steps was set to 10000
- include early stopping to reduce runtime. Tolerance was set to 0.00001 and patience to 100.
- provide a simple object to simply the following:
  - (re)fit a set of triples based on a set of meta-parameters for all variables of interest. This simply calls the code from `https://github.com/charliemarx/disentangling-influence`.
  - computation of the influence value. This calls the code from `https://github.com/charliemarx/disentangling-influence` however, following [4] the off-the-shelf use of SHAP as the underpinning direct influence function is replaced by unconditional permutation importance. In addition the reporting of influence as either a difference or a ratio is coded.
  - a method to replicate the diagnostic graphs shown in [4] from the output is included

Dependencies, download links and install instructions are included in the notebooks in the case a local install is desired. If running the notebooks on a hosted instance this will be automatically installed.

2.1.4 Source code for a modified version of the R package party

A modified version of `https://cran.r-project.org/web/packages/party/index.html` as an installable package has been made. The modification is minor, slightly altering the varimp function (in varimp.R) to, on request, return the conditional permutation importance as a ratio instead of a difference. The party version modified was 1.3-4.

Dependencies, download links and install instructions are included in the notebooks in the case a local install is desired. If running the notebooks on a hosted instance this will be automatically installed.

2.2 Synthetic Experiments

The following implementation points are of note beyond those detailed in the paper:

- The reference model for the RF MCR is a model trained as required by the MCR method (no bootstrapping, a small number of variables considered at each split, in this case 1 given there are only three variables). The model is able to learn the relationship perfectly.
- Conditional Variable importance (Random Forest version) used the implementation from the R package party `https://rdrr.io/cran/party/man/varimp.html`. Default parameters were used.
- The disentanglement (INFL) method had it's meta-parameters optimised via the previously discussed grid search method. This learns an optimised model per input variable. Despite this significant tuning (and some further manual tuning attempts), the method was relatively unstable leading to a number of solutions depending on the different random seed used. Figure 1 shows the the diagnostic graphs as considered in [4]. They indicate that the resulting networks are relatively acceptable for the run reported in the paper. With respect to interpreting these graphs, [4, pg 6] note that: *Optimal is reconstruction error and prediction error of 0 for all features (indicating no errors in autoencoding), and disentanglement error of 1 for all features.* We note that some of the alternative solutions for different seeds observed corresponded to a slightly better performance of the method in the context of the paper, though with worse diagnostic graphs. In all cases the discussion regarding this method in the paper holds regardless of the exact observed result during experimentation. Note that the notebook does not have a fixed seed and this instability can be explored by re-running the notebook.
- SHAP values are calculated on an identical RandomForestClassifier as used for the RF MCR.
- Unconditional and exact conditional permutation importance is computed on the same RandomForestClassifier as RF MCR.

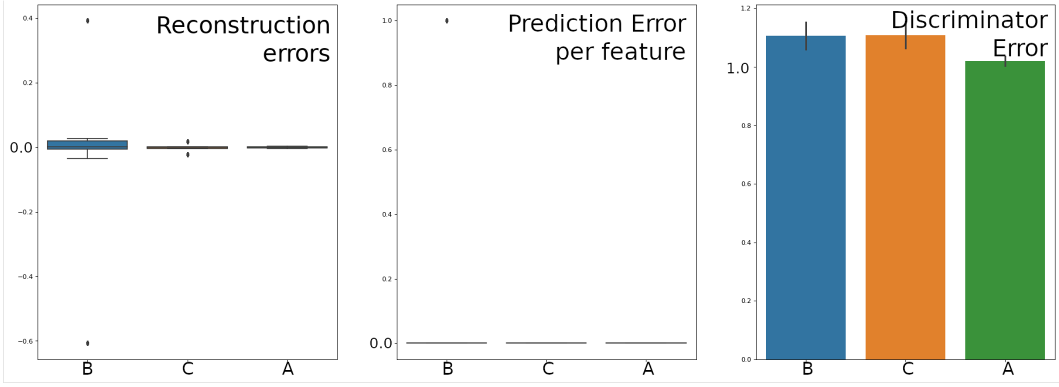


Figure 1: Graph of diagnostic measures for the neural networks trained as part of the Disentangling Influence (INFL) method from [4] for the Synthetic experiments. Graphs indicate the networks are reasonably well trained according to [4].

2.3 COMPAS Experiments

The following implementation points are of note beyond those detailed in the paper:

- The data set was the same as used in [1] and was downloaded from `https://github.com/aaronjfisher/mcr-supplement`. The training and test sets as defined by [1] were used. The notebook file details the instructions to convert the data from that repository to csv for use in Python.

- Reference models were trained (included cross-validated meta-parameter search when applicable) using the training data. Reported model performance scores for the reference models were are for the test data set.
- In this work we consider the in-sample estimation of the MCR. While in-sample estimation is motivated in [1] the reported figures for grouped admissible and inadmissible variables in their evaluation based on the COMPAS data is based on sample-splitting. In order to verify the RBF-SVM implementation an exact replica of their experiment was run which confirmed the results they reported.
- The graphs generated by the Notebooks are per MCR estimation method, rather than the comparison graphs shown in the paper. This format was generated separately from the outputs. The information is the same.
- For RF MCR a Grid Search was performed to determine the optimal meta-parameters underpinned by 5-Fold cross-validation based on the training set. The reported MSE was computed on a held out test set. The grid search considered five different levels of minimum decrease in impurity and two different values for the number of features to be considered in each split (1 and the square root of the number of input features). Bootstrapping was disabled.
- For RBF-SVM the optimal meta-parameters found within the evaluation on this dataset by [1] were used. This was based on a meta-parameter search as per [1].
- The disentangling influence method utilized the same Random Forest reference model as RF-MCR. When learning the neural network as part of the variable importance procedure, the meta-parameters were set via the grid search method previously detailed. Diagnostic graphs for this process are shown in Figure 2. They indicate that the neural networks have done a reasonable, but imperfect, job of learning a disentangled representation. This provides some explanation as to the differing performance of this hybrid method compared to RF MCR+ and RBF-SVM MCR+. The error profile is not too dissimilar to those reported in the original authors work [4] in their empirical evaluations (on different datasets), and it is unclear if it is possible to learn a better representation. As discussed in the main paper, the requirement and time needed to tune this approach additionally makes its re-purposing to the use as an MCR+ measure less desirable (grid search took a several hours despite utilization of 10 cores). Further, as discussed in the section RF-MCR Analysis in the paper and below, the requirement to learn neural networks (even with fixed meta-parameters) as part of generating importance values results in run-times significantly greater than RF-MCR and RBF-SVM MCR. For COMPAS this took 20 minutes. In contrast RF-MCR runs in < 5 seconds. Note that in both the disentanglement method and RF MCR used the same trained reference model in the experiments.

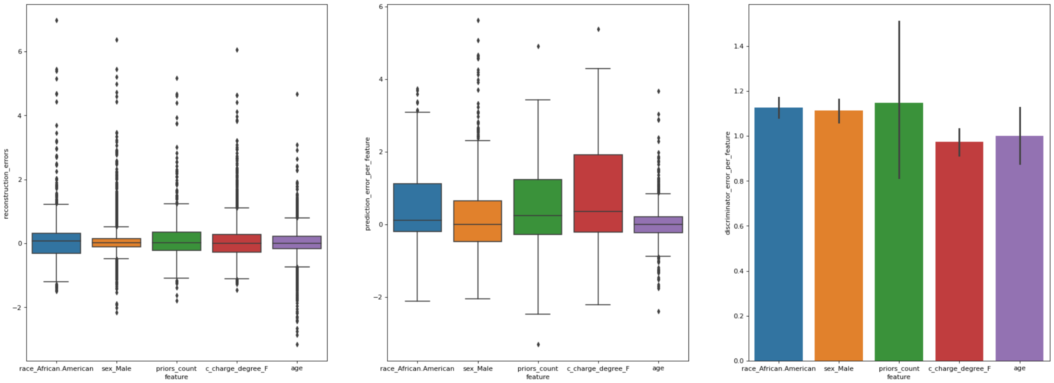


Figure 2: Graph of diagnostic measures for the neural networks trained as part of the Disentangling Influence (INFL) method from [4] for the COMPAS experiments.

2.4 Breast Cancer Experiments

The following implementation points are of note beyond those detailed in the paper:

- The data set used was accessed directly from sklearn, via a corresponding API call (sklearn version 0.23.1). In order to reduce the number of variables, and allow clearer interpretation of results, a lightweight variable selection process was applied. This took the form of a traditional approach based on unconditional variable importance for a Random Forest re-run many times with different random seeds. This process took place *before* any methods reported in this work were considered. Input variables retained were: mean compactness, mean symmetry, area error, compactness error, symmetry error, worst radius, worst texture, worst perimeter, worst compactness, worst concave points and worst symmetry. The data was then randomly split into a training (66%) and test set (33%).
- Reference models were trained (included cross-validated meta-parameter search when applicable) using the training data. Reported model performance scores for the reference models were using the test data set, and achieved an accuracy of 95.74%.
- The reference model used in both RF-MCR and for the disentanglement influence method as part of the Hybrid-VI MCR was the same. An identical grid search for the forest meta-parameters was conducted as done for the COMPAS experiment. For conditional permutation importance (MCR- as part of the Hybrid-VI MCR) a new Random Forest is required to be trained in the R environment.
- As with the COMPAS experiments we report the in-sample estimation of the MCR.
- The disentangling influence method from [4], which is re-purposed to act as the MCR+ for the Hybrid-VI method, learnt the neural network's meta-parameters via the previously described grid search considering 80 different meta-parameterizations. The diagnostic graphs are shown in Figure 3. The graphs again indicate that although the networks have learnt relatively well, there remains per-feature error differences. Since these independently affect each variable's importance, it casts some doubts over the relative variable importances implied by this method, as discussed in the paper.
- The graphs generated in the Notebooks are per-method (rather than the comparison graphs collated and illustrated in the paper). This format was generated separately from the outputs. The information is the same.

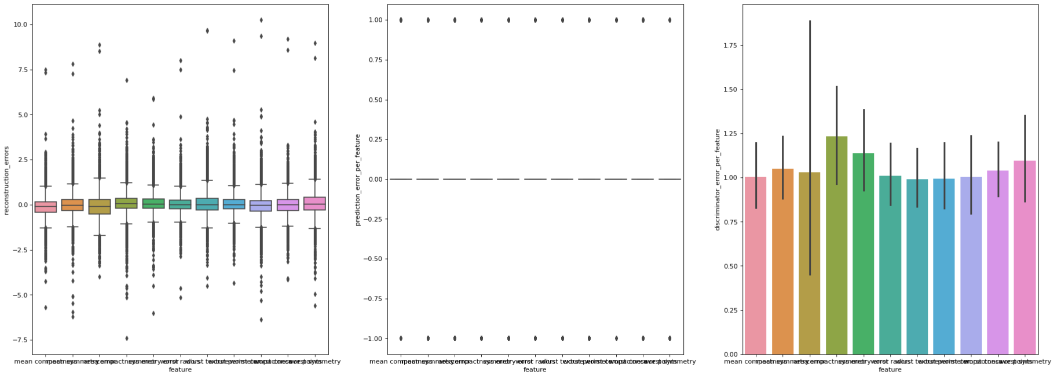


Figure 3: Graph of diagnostic measures for the neural networks trained as part of the Disentangling Influence (INFL) method from [4] for the Cancer experiments. Variables are listed along the x-axis.

2.5 RF-MCR Analysis

Runtime analysis of RBF-SVM MCR and the proposed RF MCR with respect to the number of data points was conducted on the aforementioned machine with an Intel® Core^{TM} i7-3930K CPU @ 3.20GHz with 64GB RAM and running Ubuntu 20.04, Python 3.6.10 and R 3.6.3. Both methods used single core algorithms. A comparison to the Hybrid-VI method was not made due to the method requiring the training of neural networks (and additionally the time to learn optimal meta-parameters for these networks) which meant the approach ran orders of magnitude slower (20mins vs. <5 seconds for a preliminary run on COMPAS). The runtime performance considered only the computation of the MCR (both MCR- and MCR+) and excluded the time taken to train the reference model in both cases, however, the reference model was retrained at each iteration. This is because the runtime of the MCR algorithms are somewhat dependent on the complexity of the reference model structure. Simply increasing the dataset on which the MCR is computed would not be an accurate reflection of runtime in practice if a model trained on a smaller dataset was considered during a timing study. As such we consider the computation of in-sample MCR where the training dataset and the data set used to compute the MCR is of the same size at each iteration. A final note is that the two method differ slightly in the permutation scheme used. RBF-SVM MCR is coded in the implementation by [1] to use e-divide (Equation 3.4, pg. 9). In contrast in RF MCR we use repeated random permutations, repeating 10 times. The latter is more computationally expensive meaning the proposed method, RF MCR, is slightly disadvantaged. Given the results and the common place usage of repeated random permutations e-switch was not implemented and compared to within RF MCR.

The convergence of MCR+/- to a fixed point was additionally investigated on both the COMPAS and Breast Cancer data sets. For the COMPAS dataset an ϵ value of 3.1713 was used. To show convergence under different ϵ, $\epsilon = 0.00001$ was used for the Breast Cancer dataset. In the paper, due to space constraints only two of the four graphs were presented, the convergence of MCR- for COMPAS and MCR+ for Cancer. The full four graphs are shown in Figures 4 - 7. Note, due to a different random seed being used during tree construction, the graphs are differ slightly to those in the full paper. The graphs show the same information, however, that quick convergence to a point of stability is achieved providing further evidence this occurs generally. Given the slight difference (exaggerated due to the change in scale) more trees may be justifiable for the COMPAS dataset, which might then enable the approach to report a slightly lower MCR- for age.

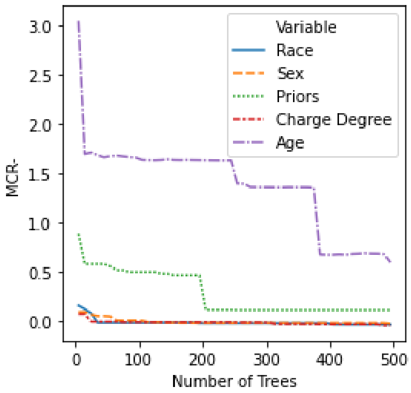


Figure 4: COMPAS dataset MCR- Results

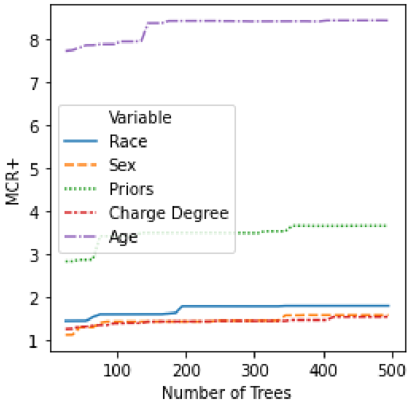


Figure 5: COMPAS dataset MCR+ Results

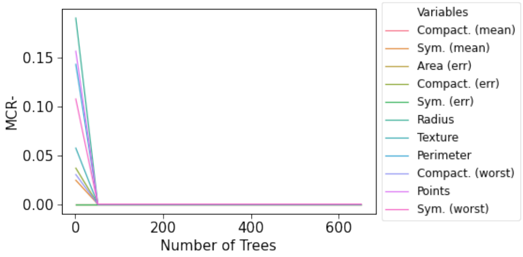


Figure 6: Breast Cancer dataset MCR- results

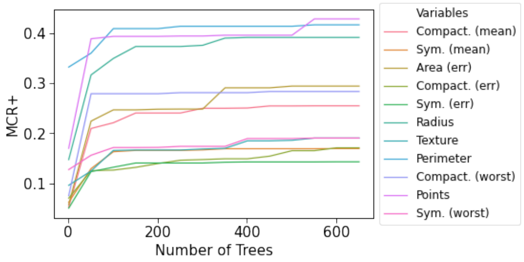


Figure 7: Breast Cancer dataset MCR+ results

Finally an ablation study was conducted, in order to investigate the extent to which the two steps of the approach (surrogate replacement and the use of prediction equivalent trees, the T^+ transform) contribute to final estimation of MCR bounds. Specifically, we examined how much each step either increases (MCR+) or decreases (MCR-) the importance (from the reference model value) per variable on both the COMPAS and Cancer datasets. Note that, for reasons as listed in Section 4.3 in the paper, in the current implementation the epsilon parameter for surrogates is not implemented. The effect of the surrogate transform, therefore, is likely to under-report, particularly for larger ϵ values. To limit this effect ϵ was set to 0.1 for the ablation study. Results of the ablation study are shown in Figures

8 and 9, with both components evidenced as playing a notable role in determining the MCR- and MCR+ values.

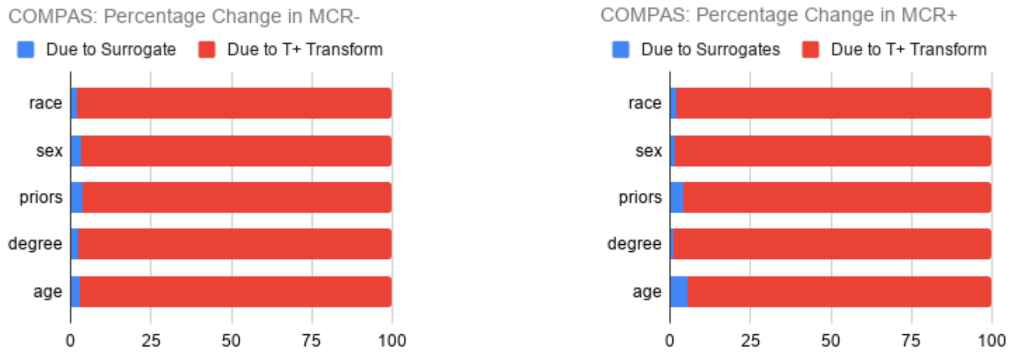


Figure 8: Ablation Study results for the COMPAS dataset

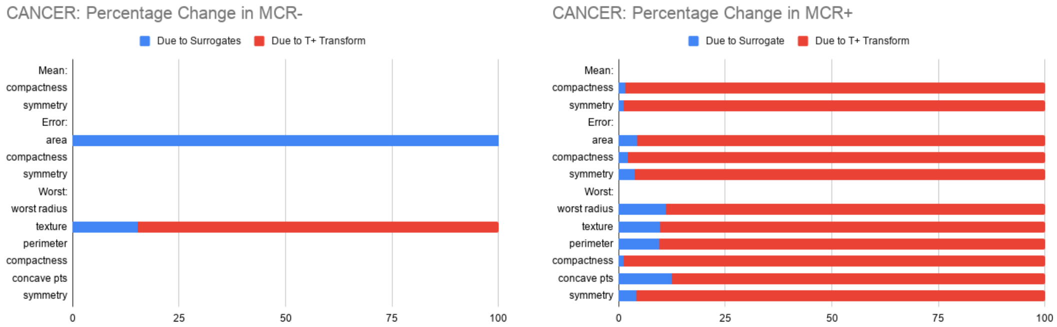


Figure 9: Ablation Study results for the Breast Cancer dataset. For variables where the reference model's importance equalled the MCR- value no increase occurred and so no percentage change is shown.

3 Algorithm Reliance and Model Class Reliance

Algorithm reliance measures the predictive performance lost when training a model without one or more variables, compared to the predictive performance of a model trained with all variables included. By definition this is not a measure of MCR (see [1, §3.2]). [1, §9.1] consider two types of AR, *necessity* and *sufficiency*. Necessity is computed by training a model with all features (M0) and a second with all features except the feature of interest (M1). The difference in predictive performance is then measured. Sufficiency is computed by again considering M0, but also considering a model trained just on the variable of interest (M2). As [1] show empirically, AR necessity lower bounds MCR-. AR sufficiency, however, does not provide a good upper bound for MCR+, failing to capture and acknowledge the variables importance within interactions with other variables. Computing AR necessity and sufficiency for both the COMPAS and Cancer datasets show results inline with those reported by [1] when comparing AR and MCR. For COMPAS, AR necessity again lower bounds reported MCR- (trivially for charge degree, sex and race while for Age and priors count ratios of 1.66 and 1.074 are reported). For the Cancer dataset, since MCR- already identifies all variables as within some model not being required, the MCR- and AR are equivalent. AR sufficiency was also computed and was always less than the MCR+ bound as expected, highlighting its inability to attribute interaction effects to the variable.

4 Additional Empirical Analysis: Predicting negative birth outcomes

As part of the empirical analysis, our proposed methodology was also applied to a third dataset, with a focus on the application of method to a pressing real-world problem, rather than comparison to baselines. The data set selected is from an ongoing project in East Africa looking at predicting negative birth outcomes based on surveyed data from: (1) community health worker visits taken before a birth, (2) living conditions and (3) features aggregated to a region level from alternative data sources such as mobile money records and call detail records. The output feature was a binary encoded variable indicating if the birth was subsequently a negative one or not.

The goal was to understand: (1) the predictive performance that could be obtained from modelling this issue (2) the potential utility of using (easier to collect) area-level and living condition features, under the assumption that the three feature types would likely share predictive information regarding the output variable. This information was then used as a basis for a real-world review into the type of information collected by the community health workers and as as basis for further future feature engineering in order to better develop the predictive model. Figure 10 shows the results of the RF-MCR analysis based on a reference model that achieved 66.5% accuracy on a held out test set.

In contrast to what would be seen by a traditional (e.g. permutation) variable importance analysis we see that region level features (home_ct, avg_trans_sent, avg_contacts, avg_sent_overall_trans, avg_trans_received) can range in importance, indicating well performing models exist where these features are utilized. This was equally true of the living condition features. Equally it showed that a number of survey questions did not seem to aid the prediction, with their almost zero MCR+ scores indicating they could (at least for this task) be considered for removal from the survey. The remaining survey questions (age, driver arranged in advance, partner permission, time of year of first visit, abortions) however show relatively large MCR- scores. These should therefore not be considered for exclusion from future surveying, as our MCR analysis indicates that they are indeed required in all models. An exception is perhaps the requirement to collect information on abortions (previously viewed as crucial), which might be revisited given the sensitive nature of the topic and the limited predictive impact indicated. Finally, results indicate that the derived features and living condition features share information about the output.

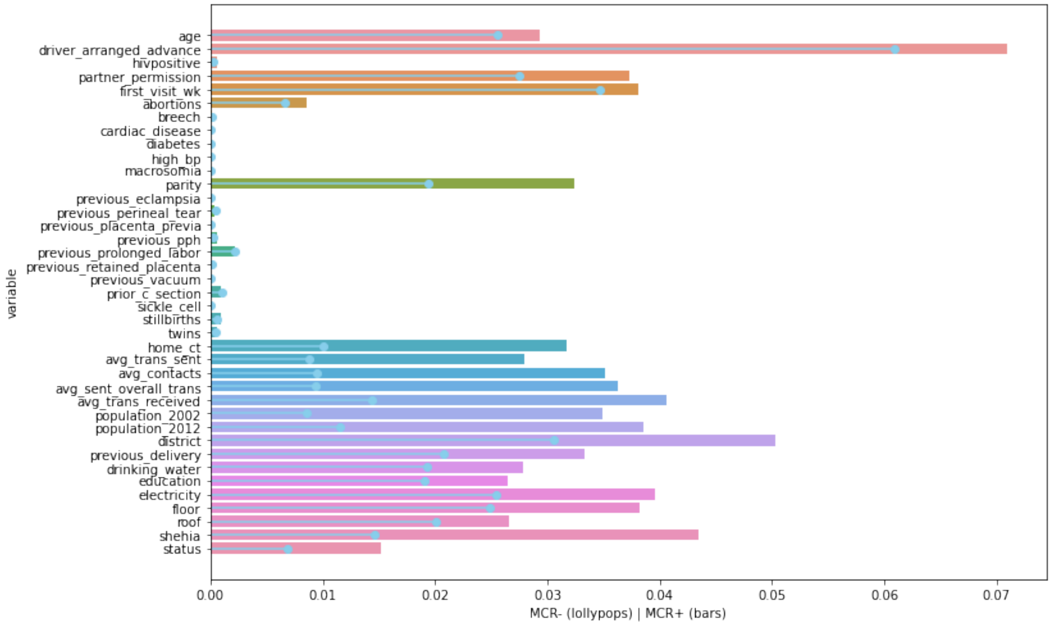


Figure 10: Negative Birth Outcomes dataset MCR Results

5 Proof

This section provides proof of the claim that:

$$MR_{X_1}(F_T^+, \Phi) \geq MR_{X_1}(F_S^+, \Phi) \tag{1}$$

Here $MR(.)$ is the model reliance of some random forest, F, on variable, X_1 given the context of $Z = \langle X_1, X_2, Y \rangle$. X_2 reflects the set of independent variables used in addition to X_1 by the forest to predict target variable, Y. Calculating MR(.) requires specification of a disruption function, Φ, that is applied to Z, to generate a version of the variable interest, $X_1^\Phi \in Z^\Phi$, that is rendered as uninformative as possible in relation to Y. Finally, we recall that model reliance is defined as:

$$\begin{aligned} MR_{X_1}(F, \Phi) &= L(F, Z^\Phi) - L(F, Z) & (2) \\ &= L(F, \langle X_1^\Phi, X_2, Y \rangle) - L(F, \langle X_1, X_2, Y \rangle) & (3) \end{aligned}$$

Below we show that following transformation of model F_S^+ to model F_T^+ (as per equation [9] in the full paper) model reliance on variable X_1 must be greater than or equal to its previous level.

We note for the proof that the transformation to F_T^+ is constructed so as to ensure *predictive equivalence* has be maintained. Specifically:

> *[The construction of F_T^+] is achieved by a second transformation $\mathcal{T}^+(F_S^+)$, mapping each tree in F_S^+ to some other in F_S^+ that produces the same predictions but which leverages our variable of interest the most (n.b. if a tree is already the most reliant on X_1 it is mapped to itself).*

Formally this means that, by construction:

$$\begin{aligned} \forall x \in Z, f \in F_S^+ :\ f(x) = \mathcal{T}(f(x)) & \quad (4) \\ \implies \forall x \in Z :\ F_S^+(x) = F_T^+(x) & \quad (5) \\ \implies L(F_S^+, Z) = L(F_T^+, Z) & \quad (6) \end{aligned}$$

Given these preliminaries, we first we substitute equation 3 into equation 1 to yield:

$$\begin{aligned} L(F_T^+, Z^\Phi) - L(F_T^+, Z) \geq L(F_S^+, Z^\Phi) - L(F_S^+, Z) & \quad (7) \\ \implies L(F_T^+, Z^\Phi) \geq L(F_S^+, Z^\Phi) & \quad (8) \end{aligned}$$

Assuming that each F reflects a random forest regressor, and that we are using a squared loss function to evaluate model reliance, then:

$$\implies \sum_{x^\Phi, y \in Z^\Phi} \left((\frac{1}{|F_T^+|} \sum_{f_T \in F_T^+} f_T(x^\Phi)) - y \right)^2 \geq \sum_{x^\Phi, y \in Z^\Phi} \left((\frac{1}{|F_S^+|} \sum_{f_S \in F_S^+} f_S(x^\Phi)) - y \right)^2 \tag{9}$$

Note that both the LHS and the RHS of this inequality now corresponds the total error of a Random Forest predictor against dataset Z^Φ. This observation, however, allows us to leverage the *ambiguity decomposition* derived in [2], which guarantees the generalization error of each ensemble to be lower than the average generalization error of its constituents. Formally the decomposition states that for any random forest, F,:

$$L(F, Z) = E(F, Z) - A(F, Z) \tag{10}$$

where:

$$E(F, Z) = \frac{1}{|F|} \sum_{f \in F} L(f, Z) \tag{11}$$

$$A(F, Z) = \mathbb{E}_{\mathbb{Z}} \left[\frac{1}{|F|} \sum_{f \in F} (f(x) - F^+(x))^2 \right] \tag{12}$$

$E(.)$ corresponds to the average generalization error of the individual trees in the forest, while the second term is the *ensemble ambiguity*, reflecting the variance of individual predictions made by the forest as a whole. Since A(.) is non-negative, the generalization error of the ensemble is therefore always smaller than equal to the average generalization error of its constituents [3][pg. 63].

We applying the ambiguity decomposition directly to both sides of equation 9:

$$\left[\frac{1}{|Z|}\sum_{x^\Phi,y\in Z^\Phi}\frac{1}{|F_T^+|}\sum_{f_T}(f_T(x^\Phi)-y)^2\right]-\left[\frac{1}{|Z|}\sum_{x^\Phi,y\in Z^\Phi}\frac{1}{|F_T^+|}\sum_{f_T\in F_T}\left(f_T(x^\Phi)-F^+(x^\Phi)\right)^2\right]$$
$$\geq$$
$$\left[\frac{1}{|Z|}\sum_{x^\Phi,y\in Z^\Phi}\frac{1}{|F_S^+|}\sum_{f_S}(f_S(x^\Phi)-y)^2\right]-\left[\frac{1}{|Z|}\sum_{x^\Phi,y\in Z^\Phi}\frac{1}{|F_S^+|}\sum_{f_S\in F_S}\left(f_S(x^\Phi)-S^+(x^\Phi)\right)^2\right] \tag{13}$$

Recognizing that the cardinality of the model doesn't change following transformation, such that $|F_T^+| = |F_S^+|$, and rearranging gives us:

$$\left[\sum_{x^\Phi,y\in Z^\Phi}\sum_{f_T\in F_T^+}(f_T(x^\Phi)-y)^2\right]-\left[\sum_{x^\Phi,y\in Z^\Phi}\sum_{f_S\in F_S^+}(f_S(x^\Phi)-y)^2\right]\geq$$
$$\left[\sum_{x^\Phi,y\in Z^\Phi}\sum_{f_T\in F_T^+}\left(f_T(x^\Phi)-F_T^+(x^\Phi)\right)^2\right]-\left[\sum_{x^\Phi,y\in Z^\Phi}\sum_{f_S\in F_S}\left(f_S(x^\Phi)-F_S^+(x^\Phi)\right)^2\right] \tag{14}$$

This equation must always hold. This can be proven by considering the behaviour of each term on the left hand side of the equation (*LHS:T1*, *LHS:T2*) and right hand side (referred to as *RHS:T1*, *RHS:T2*), in light of potential bijective mappings of F_T^+ as it is transformed from F_S^+. Consider first the degenerative case where the transformation $\mathcal{T}^+ : F_S^+ \to F_S^+$, maps every element to itself so F_T^+ = F_S^+. In this case both terms on the LHS and RHS of equation 14 would be equivalent, both sides resolving to zero.

However, if the mapping $\mathcal{T}^+$ finds a tree that is substitutable for another in $F{+}_S$ (such a tree being predictively equivalent yet of greater model reliance), then while both *LHS:T2* and *RHS:T2* stay the same, both *LHS:T1* and *RHS:T1* must increase. This is guaranteed by construction, as the substitution undertaken by $\mathcal{T}^+$ only occurs if squared error against Z^Φ) is increased (the definition of increased model reliance). Consequently equation 14 holds if and only if the resulting change in *LHS:T1* is greater than or equal to the *RHS:T1*:

$$\Delta\left[\sum_{x^\Phi,y\in Z^\Phi}\sum_{f_T\in F_T^+}(f_T(x^\Phi)-y)^2\right]\geq\Delta\left[\sum_{x^\Phi,y\in Z^\Phi}\sum_{f_T\in F_T^+}\left(f_T(x^\Phi)-F_T^+(x^\Phi)\right)^2\right] \tag{15}$$

To prove this is always the case, consider that both components are of the form: $\sum_{f_S\in F_T^+}(f_T(x)-q)^2$. Minimizing this equation across all datapoints, given q is a constant, gives us:

$$\frac{d}{dq}\left(\sum_{f\in F_T}(f(x)-q)^2\right)=0 \implies \tilde{q}=\frac{1}{|F|}f(x)=F_T^+(x)$$

Thus the right hand side term of equation 15 is minimized for any given, Z^Φ - and as a consequence the left hand side term can only be greater or equal to it. Therefore equation 15 holds, as does equation 14, and consequently also equation 8. Thus:

$$MR_{X_1}(F_T^+,\Phi)\geq MR_{X_1}(F_S^+,\Phi) \tag{16}$$

This completes the proof.

References

[1] Aaron Fisher, Cynthia Rudin, and Francesca Dominici. All models are wrong, but many are useful: Learning a variable's importance by studying an entire class of prediction models simultaneously. *Journal of Machine Learning Research*, 20(177):1–81, 2019.

[2] Anders Krogh and Jesper Vedelsby. Neural network ensembles, cross validation, and active learning. In *Advances in neural information processing systems*, pages 231–238, 1995.

[3] Gilles Louppe. *Understanding Random Forests: From Theory to Practice*. PhD thesis, Université de Liège,Liège, Belgique, 2014. URL `https://arxiv.org/pdf/1407.7502.pdf`.

[4] Charles Marx, Richard Phillips, Sorelle Friedler, Carlos Scheidegger, and Suresh Venkatasubramanian. Disentangling influence: Using disentangled representations to audit model predictions. In *Advances in Neural Information Processing Systems*, pages 4498–4508, 2019.