
Detecting Statistical Interactions with Additive Groves of Trees

Daria Sorokina
Rich Caruana
Mirek Riedewald

Department of Computer Science, Cornell University, Ithaca, NY, USA

Daniel Fink

Cornell Lab of Ornithology, Ithaca, NY, USA

DARIA@CS.CORNELL.EDU
CARUANA@CS.CORNELL.EDU
MIREK@CS.CORNELL.EDU

DF36@CORNELL.EDU

Abstract

Discovering additive structure is an important step towards understanding a complex multi-dimensional function because it allows the function to be expressed as the sum of lower-dimensional components. When variables interact, however, their effects are not additive and must be modeled and interpreted simultaneously. We present a new approach for the problem of interaction detection. Our method is based on comparing the performance of unrestricted and restricted prediction models, where restricted models are prevented from modeling an interaction in question. We show that an additive model-based regression ensemble, Additive Groves, can be restricted appropriately for use with this framework, and thus has the right properties for accurately detecting variable interactions.

1. Introduction

Many scientific inquiries seek to identify what variables are important and to describe their effects. Discovery of additive structure is an important step towards understanding a complex multi-dimensional function, because it allows for expressing this function as the sum of lower-dimensional components. When variables interact, their effects cannot be decomposed into independent lower-dimensional contributions and hence must be modeled simultaneously. In this paper we develop a methodology to automatically identify additive and interactive structure among large sets of variables.

Appearing in *Proceedings of the 25th International Conference on Machine Learning*, Helsinki, Finland, 2008. Copyright 2008 by the author(s)/owner(s).

The term *statistical interaction* is used to describe the presence of non-additive effects among two or more variables in a function. Two variables are said to interact when the effect of one variable on the response depends on values of the other variable. Precisely, variables x_i and x_j interact in $F(\mathbf{x})$ when partial derivative $\frac{\partial F(\mathbf{x})}{\partial x_i}$ depends on x_j or, more generally, when the “difference in the value of $F(\mathbf{x})$ for different values of x_i depends on the value of x_j ” (Friedman & Popescu, 2005). This is equivalent to the following definition:

Function $F(\mathbf{x})$, where $\mathbf{x} = (x_1, x_2, \dots, x_n)$, shows no interaction between variables x_i and x_j if it can be expressed as the sum of two functions, $f_{\setminus j}$ and $f_{\setminus i}$, where $f_{\setminus j}$ does not depend on x_j and $f_{\setminus i}$ does not depend on x_i :

$$F(\mathbf{x}) = f_{\setminus j}(x_1, \dots, x_{j-1}, x_{j+1}, \dots, x_n) + f_{\setminus i}(x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n) \quad (1)$$

For example, $F(x_1, x_2, x_3) = \sin(x_1 + x_2) + x_1 x_3$ has interactions between x_1 and x_2 and also between x_1 and x_3 , but no interaction between x_2 and x_3 .¹

Higher-order interactions between a larger number of variables are defined similarly. There is no K -way interaction between K variables in the function, if it can be represented as a sum of K (or fewer) functions, each of which does not depend on at least one variable in question. If such representation is not possible, we say that there is a K -way interaction. Function $x_1^{x_2+x_3}$ shows a 3-way interaction between x_1 , x_2 and x_3 , while $x_1 x_2 + x_2 x_3 + x_1 x_3$ has all pairwise interactions, but not a 3-way interaction.

¹It is important to stress that the concept of statistical interaction is completely unrelated to the dependence and independence of variable distributions. Some authors use “interaction” to refer to different types of dependencies between variables, e.g., correlation (Jakulin & Bratko, 2004). In this paper we discuss statistical (non-additive) interactions only, not correlation or statistical dependence.

Interaction detection has high practical importance because it provides valuable knowledge about a domain. For example, our experiments with bird abundance data (Section 7) demonstrate that detection of spatio-temporal interactions can signal changes in the environment. In this particular case, a fatal eye disease was spreading slowly from the Northeastern US to other regions. This disease affected the annual bird abundance differently depending on location, creating a strong interaction between time and location.

Interactions are also an important part of statistical analysis. Early methods for interaction detection were parametric and required explicit modeling of interactions, most often as multiplicative terms. As a consequence, only limited types of interactions could be detected. More general approaches were introduced recently (Friedman & Popescu, 2005; Hooker, 2007). These methods are based on building a model and detecting interactions in the function learned by the model. A major shortcoming of this approach is that the model may detect spurious interactions over regions of the input space where data is scarce, and known solutions to this problem are either inadequate or computationally expensive. (See (Hooker, 2007) and Section 8 of this paper for more details.)

We introduce a new approach to interaction detection. It is based on comparing the performance of *restricted* and *unrestricted* predictive models. This avoids the drawbacks of previous methods, because it does not require explicit modeling of interacting terms and reports only those interactions that are present in the actual input data. However, the choice of model and the restriction algorithm used are crucial for this framework. We explain why additive models are able to provide the required accurate restrictions and further show that Additive Groves (Sorokina et al., 2007), an additive model-based ensemble of regression trees, works well in this framework. We also investigate how correlations in the data complicate interaction detection and suggest how this problem can be dealt with via feature selection.

The advantage of our new approach for interaction detection, compared with traditional statistical approaches, is that it is more automatic and does not require limiting the functional form that interactions might take. Statistical methods often represent only multiplicative interactions and thus may miss other forms of interactions. When little is known about the system under study, data-driven scientific discovery requires the data to “speak for themselves” with a minimum of analyst input or assumptions. It is possible to conduct a fully nonparametric analysis with the

method we propose in this paper, which is particularly valuable for exploratory analysis.

2. Estimating Interactions

Let $F^*(\mathbf{x})$ be an unknown target function and let $F(\mathbf{x})$ be a highly accurate model of F^* that can be learned from a given set of training data. Furthermore, let $R_{ij}(\mathbf{x})$ denote a *restricted* model of F^* that is learned from the same training data. It is restricted in the sense that it is not allowed to contain an interaction between x_i and x_j , but apart from this limitation should be as accurate a model of F^* as possible.

Our interaction estimation technique is based on the following observation. If x_i and x_j interact, then $F(\mathbf{x})$ should have significantly better predictive performance than $R_{ij}(\mathbf{x})$, because the latter cannot accurately capture the true functional dependency between x_i and x_j . On the other hand, if the two variables do not interact, then the absence of the interaction from the model should not hurt its quality. Hence in the absence of an interaction between x_i and x_j the predictive performance of the restricted and the unrestricted model should be comparable. Note that in order to get an adequate estimate of performance, we must measure it on test data not used for training.

Quantifying interaction strength. We can quantify I_{ij} , the degree of interaction between x_i and x_j , by the difference in performance between $F(\mathbf{x})$ and $R_{ij}(\mathbf{x})$. We measure performance as standardized RMSE: root mean squared error (RMSE) scaled by the standard deviation in the response function. Scaling is done to make the results comparable across different data sets; $\text{StD}(F^*(\mathbf{x}))$ is calculated as standard deviation of the response values in the training data.

$$\text{stRMSE}(F(\mathbf{x})) = \frac{\text{RMSE}(F(\mathbf{x}))}{\text{StD}(F^*(\mathbf{x}))} \quad (2)$$

$$I_{ij}(F(\mathbf{x})) = \text{stRMSE}(F(\mathbf{x})) - \text{stRMSE}(R_{ij}(\mathbf{x})) \quad (3)$$

Setting the threshold. To distinguish whether a positive value of I_{ij} indicates presence of an interaction or happened due to random variation, we measure whether the performance of $R_{ij}(\mathbf{x})$ is significantly different from the performance of $F(\mathbf{x})$. We follow common practice and define a difference of three standard deviations of the latter from its mean as significant. The distribution of $\text{stRMSE}(F(\mathbf{x}))$ can come either from different random seeds for bagging or from different data samples (e.g., n -fold cross validation). The threshold for significant interactions then becomes:

$$I_{ij}(F(\mathbf{x})) > 3 \cdot \text{StD}(\text{stRMSE}(F(\mathbf{x}))) \quad (4)$$

Note that everything above naturally generalizes to higher-order interactions as long as there exists a method to restrict the model on a specific type of interaction.

3. Choosing a Prediction Model

To correctly estimate interaction strength with our model comparison technique, we have to make sure that a model has the following key properties:

1. High predictive performance when modeling interactions: if there is an interaction, it should be captured by the unrestricted model.
2. High predictive performance when the model is restricted on non-interacting variables: if there is no interaction, performance of the restricted model should be no worse than the performance of the corresponding unrestricted model.

The first requirement is satisfied by many learning techniques, e.g., bagged decision trees of adequate depth, SVMs, or neural nets. Boosted stumps, on the other hand, do not model interactions. Since they represent functions as the sum of components, each of which depends only on a single variable, boosted 1-level stumps cannot be used in our framework.

While many models satisfy the first requirement, the second requirement — that models perform as well when interaction between non-interacting variables is restricted — is far more challenging. Even when there is a straightforward way of explicitly preventing specific interactions, often the resulting restricted model will not perform as well as the unrestricted model because the restriction may hamper the search in model space compared to the unrestricted model.

Consider a single decision tree. Variables in the tree can interact only if they are used on the same branch of the tree. So the obvious way to restrict interaction between specific variables is to not use one of them if the other already was used earlier on this branch. Now suppose there is no interaction between variables A and B , but they both are important — if the tree does not use one of them, its performance drops. Assume further that A is more important than B . The tree will tend to choose A earlier than B on all branches (in the worst case it will use A at the root) and will then never be able to choose B . Since B is important, the performance of this restricted tree will drop even though there was no interaction between A and B .

One might be tempted to address this problem with an ensemble method like bagging. Unfortunately the situation will not improve much. In bagging, every tree tries to capture the same function from a different

sample of the train set. If A is more important, most trees will choose A before B , use of B will be restricted, and performance will drop as before.

Additive models. To detect *absence of interactions* between important variables, we need to build a restricted model that uses these variables in different additive components of the function. There is a class of ensembles that allows us to do this: additive models. Each component in an additive model is trained on the residuals of predictions of all other previous models in the ensemble. The training set for the new model component is created as the difference between true function values and current predictions of the ensemble. This way, when the function has additive structure, different models (or groups of models) are forced to find and model different components of this structure as opposed to each modeling the whole function.

Not all models that fit residuals are suitable for this framework. Linear models do not model interactions, while generalized linear models disguise additive structure with a non-linear transformation. Neural networks pose problems because they either have additive structure (1 internal layer), or the ability to model complex non-linear functions (several layers), while we need an algorithm that combines both. Restricting interactions in a multi-level network splits it into sub-nets, ultimately leading to "groves of nets".

In this paper we use layered Additive Groves (Sorokina et al., 2007). There exist other methods that might work as well, e.g., gradient boosting trained to minimize least squares loss (Friedman, 2001). However, it is important to understand that the two requirements stated in the beginning of this section are crucial and many (most?) learning algorithms do not satisfy them.

4. Additive Groves of Regression Trees

Additive Groves is an ensemble of trees introduced in (Sorokina et al., 2007). The combination of the ability to model additive structure of the response and to also use large trees that capture complex interactions make Groves suitable for interaction detection.

A single Grove of trees is an additive model where each additive component is represented by a regression tree. Additive Groves use regression trees trained to minimize mean squared error. Tree size is controlled by a parameter α , the minimum fraction of train set cases in a non-leaf node. A single Grove is trained similar to an additive model: each tree is trained on the residuals of the sum of the predictions of the other trees. Trees are discarded and retrained in turn until the overall predictions converge to a stable function. For the pur-

Algorithm 1 Layered training of a single Grove

```

function Layered( $\alpha, N, TrainSet\{\mathbf{x}, y\}$ )
     $\alpha_0 = 0.5, \alpha_1 = 0.2, \alpha_2 = 0.1, \dots, \alpha_{\max} = \alpha$ 
    for  $i = 1$  to  $N$  do
         $Tree_i = 0$ 
    for  $j = 0$  to  $\max$  do
        repeat
            for  $i = 1$  to  $N$  do
                 $newTrainSet = \{\mathbf{x}, y - \sum_{k \neq i} Tree_k(\mathbf{x})\}$ 
                 $Tree_i = TrainTree(\alpha_j, newTrainSet)$ 
            until (change from the last iteration is small)
    
```

pose of interaction detection we use layered training of Additive Groves (Algorithm 1). The main difference between layered training and training classical additive models is the following: Additive Groves begin with an ensemble of very small trees; then during re-training we gradually increase tree size by adding more branches. This layered approach ensures fitting of additive structure of the response function. As with single trees, a single Grove can still overfit to the training data. Hence for the Additive Groves ensemble, we wrap bagging around the layered training algorithm: many single Groves are built on bootstrap samples of the training set and their results are averaged. This procedure reduces variance and yields a very powerful predictive model.

Additive models provide an intuitive and easy way for restricting interactions. Assume we want to restrict a single Grove to not contain interactions between x_i and x_j . Since the modeled function is computed as the sum of the predictions of the individual trees, we only have to enforce that none of the trees uses *both* x_i and x_j . To decide if a tree is not allowed to use x_i (or otherwise x_j), we use a greedy procedure. Each time we train a tree, we first construct two trees: one does not use x_i , the other does not use x_j . The one resulting in better performance is inserted into the model, the other one is discarded. For evaluating performance we use the out-of-bag samples, i.e., that part of the training data that did not get into the current sample and therefore was not used to train the trees.

If we need to restrict on a higher-order interaction (say, k -way interaction between k variables), we need to build k candidate trees instead of 2 every time: each tree is not allowed to use one of the variables. Note that the complexity of testing for a single k -way interaction depends only linearly on k .

(Sorokina et al., 2007) also suggest another, “dynamic programming”, style of training for Additive Groves. The method starts with a single small tree. Then on

every retraining stage it either increases tree size or adds another tree, which is decided by a heuristic. Although this method provides better performance for unrestricted models, we have encountered problems with it when training restricted models. Therefore we prefer layered Additive Groves for interaction detection. Note that we need to use layered training even for the unrestricted model in order for the performances to be comparable.

5. Feature Selection

Correlations among features are common and complicate the task of detecting interactions. Suppose there exists an interaction between variables x_i and x_j . At the same time, a third variable, x_k , is present in the data. Assume it is highly correlated with x_j , to such an extent that the model can freely use either x_k or x_j with similar results. In this case we will not be able to detect the interaction between x_i and x_j . When we restrict the model to prevent a tree from using x_j , it can use x_k instead and performance will not drop. The same will happen when we try to detect an interaction between x_i and x_k .

Correlation among features is an intrinsic problem of high dimensional data that confronts all methods for interaction detection. For example, methods based on partial dependence functions (Friedman & Popescu, 2005) suffer from a similar problem. The unrestricted prediction model might sometimes use x_j and sometimes x_k . As a result it will find only weak interaction between x_i and x_j and also between x_i and x_k , even though the true interactions are much stronger. If there are more than two correlated variables (again, this is common in high-dimensional datasets), the interaction can be spread out in tiny portions over all of them, making it virtually impossible to detect.

As a consequence, before attempting to detect interactions, we must eliminate correlations. This can be achieved by a feature selection process, which removes some of the variables. The final set of variables should be a compromise between two goals: (1) The performance of the unrestricted model should still be good, ideally at least as good as before feature selection. (2) Each variable should be important, i.e., if we remove it from the set of features, the performance of the unrestricted model should drop significantly. The second criterion also gives us an estimate of the maximum strength of interactions that we can detect: if the performance of the unrestricted model drops by δ when we remove x_i , then we cannot expect the performance of the best model restricted on x_i and x_j to drop by more than δ . The intuition here is that removing an

important variable is a stronger restriction than prohibiting its interactions.

We use a variant of backward elimination (Guyon & Elisseeff, 2003) for the feature selection process. The main idea is to greedily eliminate all features (variables) whose removal either improves performance or reduces performance by at most Δ compared to performance on the full-feature data set. In our experiments we estimated $d = \text{StD}(\text{RMSE}(F(\mathbf{x})))$, where $F(\mathbf{x})$ is the unrestricted model, before running feature selection and used $\Delta = 3d$.

The feature selection procedure is not stable—it depends on the order in which we test each feature. For example, if we consider two completely correlated variables x_j and x_k , we can remove x_j and leave x_k in the set of the features. Or we can do exactly the reverse, depending on which variable we tried to remove first during feature selection. If there is a strong notion of which features should stay in the data set after feature selection, i.e., if we want to test certain features for interactions, the feature selection process should be modified so that features of interest are not removed.

6. Complexity Issues

One concern about interaction detection is the need to conduct a separate test for each interaction. If we want to test for all possible interactions, in theory we need $O(n^k)$ tests, where n is the number of variables and k is the order of the interaction. However, such complexity is unlikely to be required in practice. First, the feature selection process usually leaves a relatively small set of features that makes it feasible to test all pairs for possible interactions. Second, as noted by (Hooker, 2004), interactions possess an important monotonicity property. A k -way interaction can only exist if all its corresponding $(k - 1)$ -interactions exist. This fact is a straightforward consequence from the definition of a k -way interaction. Hence after we have detected all 2-way interactions, we need to test for 3-way interactions only for those triples of variables that have all 3 pairwise interactions present, and so on. As complex interactions are rare in real datasets, in practice we usually need only few tests for higher-order interactions. Some domains do pose an exception, for example, see our experiments on the *kin8nm* dataset.

7. Experiments

We have applied our approach to both synthetic and real data sets. We can evaluate the performance of our algorithm on synthetic data because we know the true interactions; for real data we try to explain the

detected interactions based on the data set description.

In all our experiments we used 100 iterations of bagging. Apart from that, Additive Groves requires two parameters to be set: N (number of trees in a single Grove) and α (fraction of train set cases in the leaf, controls size of a single tree). We determined the best values of α and N on a validation set and reported the performance of Additive Groves with these parameters on a test set. We ran each experiment for the unrestricted model 10 times, using different random seeds and therefore different bootstrap samples for bagging. From these results we estimated the distribution of performance and then calculated the interaction threshold using Equation 4. After that we ran the experiment for each unrestricted model only once. If the resulting estimate of the interaction was above the threshold, we considered it to be evidence of an interaction. Otherwise it was considered insignificantly different from zero, indicating absence of an interaction. Notice that due to variance, in the latter case the estimate could be even negative, but should always be close to zero.

7.1. Synthetic Data.

This data set was generated by a function that was previously used in (Hooker, 2004).

$$F(x) = \pi^{x_1 x_2} \sqrt{2x_3} - \sin^{-1}(x_4) + \log(x_3 + x_5) - \frac{x_9}{x_{10}} \sqrt{\frac{x_7}{x_8}} - x_2 x_7 \quad (5)$$

Variables $x_1, x_2, x_3, x_6, x_7, x_9$ are uniformly distributed between 0.0 and 1.0 and variables x_4, x_5, x_8 and x_{10} are uniformly distributed between 0.6 and 1.0. Training, validation and test set contain 1000 points each. Best parameters were detected as $\alpha = 0.02$ and $N = 8$. Feature selection eliminated variables x_6 (not present in the function) and x_8 (virtually no influence on the response). For each of the 28 pairs of remaining variables we constructed a restricted model and compared it to the unrestricted model. Figure 1 shows the interaction value for each variable pair as computed by Equation 2. The dashed line shows the threshold. We can see a group of strong interactions high above the threshold — pairs (x_1, x_2) , (x_1, x_3) , (x_2, x_3) , (x_2, x_7) , (x_7, x_9) . All cases without interactions fall below the threshold. There are also several weak interactions in the data set: our estimate for (x_9, x_{10}) is barely above the threshold and we failed to detect interactions (x_3, x_5) and (x_7, x_{10}) . By construction, x_5 and x_{10} have a small range and their interactions are not significant. There is only one triple of variables with 3 pairwise interactions detected: (x_1, x_2, x_3) . A separate test correctly reveals that there is a 3-way interaction

between them. Note that this is the only higher-order interaction that we need to test to conclude the full analysis. The original formula has another 4-way interaction, (x_7, x_8, x_9, x_{10}) , but interactions of x_8 and x_{10} turned out to be very weak in the data, so the model did not pick them up.

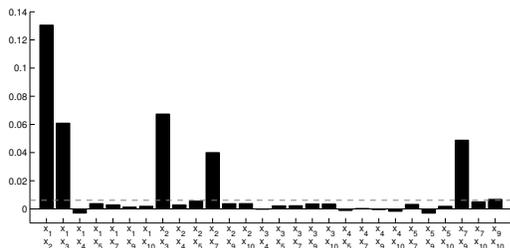


Figure 1. Interaction estimates on synthetic data

For more realistic results, we generated a version of the same data set with a 2 : 1 signal-to-noise ratio. Now feature selection left only 5 variables: x_1, x_2, x_3, x_5, x_7 , and results of interaction detection between those variables were qualitatively the same as the correspondent results for the data set without noise.

7.2. Real data sets

We have run experiments on 5 real data sets, 4 of them are regression data sets from Luís Torgo’s collection (Torgo, 2007), and the last one is a bird abundance data set from the Cornell Lab of Ornithology (Caruana et al., 2006). We used 4/5 of the data for training, 1/10 for validation and 1/10 for testing.

California Housing. California Housing is a regression data set introduced in (Pace & Barry, 1997). It describes how housing prices depend on different census data variables. Parameters used: $\alpha = 0.0005$, $N = 6$. Feature selection identified six variables as important: longitude, latitude, housingMedianAge, totalRooms, population and medianIncome. (Hooker, 2007) describes the joint effect of latitude and longitude on the response function. Our results confirm that there is a clear strong interaction between these two variables — the location effect on prices cannot be split into the sum of latitude and longitude effects. We have also found an evidence of interaction between population and totalRooms (Figure 2).

Elevators. This data set originates from an aircraft control task (Camacho, 1998). Parameters used: $\alpha = 0.02$ and $N = 18$. Feature selection left six variables: *climbRate*, *p*, *q*, *absRoll*, *diffRollRate*, *Sa*. We detected strong pairwise interactions in the triple (*absRoll*, *diffRollRate*, *Sa*) and a separate test confirmed

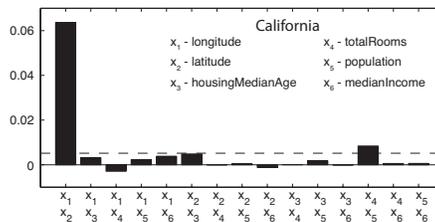


Figure 2. Interaction estimates for California Housing.

that this is indeed a strong 3-way interaction (Figure 3). No other interactions were found.

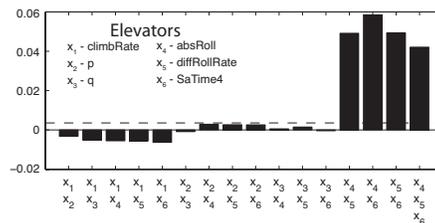


Figure 3. Interaction estimates for Elevators data.

Kinematics (kin8nm). The *kin8nm* dataset from the Delve repository (Rasmussen et al., 2003) describes a simulation of an 8-link robot arm movement. Its input variables correspond to the angular positions of the joints and it is classified as highly non-linear by its creators. Parameters used: $\alpha = 0.005$ and $N = 17$. Our analysis produced symmetrical results that reveal the simulation nature of the dataset: all 8 features turn out to be important, 2 of them do not interact with any other features and the other 6 are connected into a 6-way interaction (Figure 4). For brevity we show only results of tests for 2-way interactions and the final 6-way interaction, but we have also conducted tests for 20 3-way, 15 4-way and 6 5-way interactions between those 6 variables following the procedure described in Section 6. All tests confirmed the presence of interactions. *kin8nm* is the only data set where we had to test for many higher-order interactions. This is a property of the domain: the formula describing the end position of the arm based on joints angles results from interaction between most of the variables.

CompAct. Another dataset from the Delve repository, it describes the level of CPU activity in multiuser computer systems. Parameters used: $\alpha = 0.05$ and $N = 18$. Feature selection left 9 variables: *bread*, *scall*, *sread*, *exec*, *wchar*, *pgout*, *ppgin*, *vfft*, *freeswap*. This data set turns out to be very additive. Although there are many 2-way interactions, they all are relatively small (Figure 5). The largest interactions are (*freeswap*, *wchar*), describing the joint effect of the

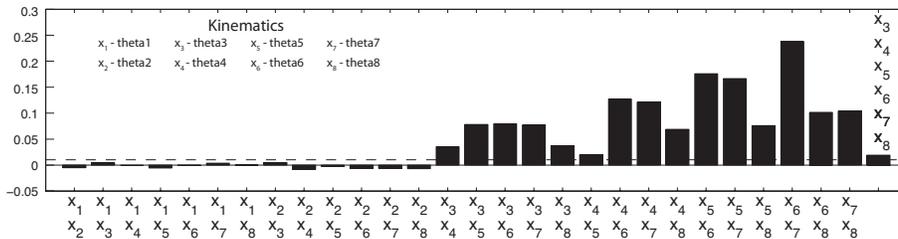


Figure 4. Interaction estimates for Kinematics (kin8nm) data.

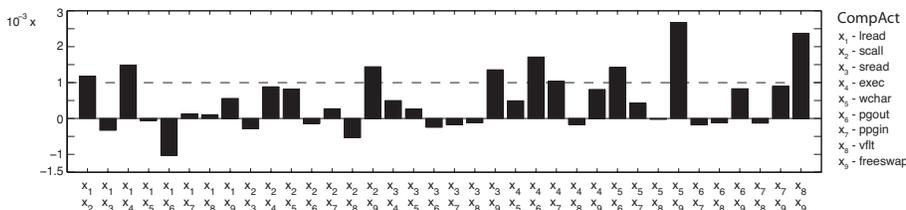


Figure 5. Interaction estimates for CPU Activity (CompAct) data set.

number of blocks available for swapping and system write call speed, and (*freeswap*, *vflt*), describing an interaction between the same available blocks variable and the number of page faults.

House Finch Abundance Data. We tested our approach on a dataset with sightings of House Finches in the North-Eastern US as introduced in (Caruana et al., 2006). The strongest interactions that we detected are between the following variables: (*latitude*, *longitude*, *elevation*) and (*year*, *latitude*, *longitude*). The first 3-way interaction describes the effect of geographical position which is expected to be non-additive. But the interactions between year and location is less trivial. Normally one would not expect that the effect of latitude or longitude on bird abundance would be very different in different years. However, it turns out that during the decade covered by the data set, the population of House Finches was suffering from an eye-disease that was spreading slowly and was responsible for changing the effect of geographical location on bird abundance over time. Our results show that interesting domain information like this can be discovered with the help of interaction detection analysis.

8. Previous Work

Interaction detection is regularly performed as part of statistical analysis (Christensen, 1996). Mostly parametric models are used where the analyst specifies the interaction as a parametric term, or perhaps several terms. In this setting interaction detection becomes a parameter estimation problem. More recently,

techniques have been developed to detect interactions within semi-parametric models (Ruppert et al., 2003).

(Friedman & Popescu, 2005) developed tests for interaction detection for a very general class of prediction models, including fully nonparametric models. Their method makes use of the fact that in the absence of an interaction between x_i and x_j the following holds: $\frac{\partial F(\mathbf{x})^2}{\partial x_i \partial x_j} = \frac{\partial F(\mathbf{x})}{\partial x_i} + \frac{\partial F(\mathbf{x})}{\partial x_j}$. They estimate the partial dependence functions (Friedman & Popescu, 2005) of the model and then estimate the strength of an interaction as the difference between the right hand side and the left hand side of the equation above, scaled by variance in the response.

The drawback of that method is that in order to get accurate estimates of the partial dependence function, it relies on predictions for synthetic data points in sparse regions of the input space. As a result, decisions about presence of interactions can be made because of spurious interactions that happen only in those regions (Hooker, 2007). To demonstrate this effect, we generated two simple data sets for the function $F(\mathbf{x}) = x_1^3 + x_2^3$. In the first data set both x_1 and x_2 are distributed uniformly between -10 and 10 . For the second data set we took the same points and removed those where both x_1 and x_2 were positive. Neither of the data sets contains interactions, but the estimates produced by Friedman’s approach using RuleFit (Friedman, 2005) were 0.0243 for the first and 0.0824 for the second set. The presence of an unpopulated region in the input data increased the estimated strength of the presumed interaction by a

factor of three.

In order to deal with this extrapolation problem, (Friedman & Popescu, 2005) suggest comparing the estimated interaction strength produced by the method described above with a similar estimate on the same data, but for a different response function that does not contain any interactions. However, our experiments with RuleFit revealed several examples of unsatisfactory performance of this technique. For instance, we generated 5 data sets with response function $x_1^2 + x_2^2$ without noise and for each of them generated 50 samples from the null distribution. For 3 of those data sets RuleFit produced results that indicated presence of an interaction, i.e., the original estimate was further from the mean of the null distribution than 3 standard deviations. In contrast, our method produced a confident estimation of the absence of interactions in all the cases described above.

(Hooker, 2007; Hooker, 2004) suggests another approach, based on estimating orthogonal components of the ANOVA decomposition. This method has higher computational complexity because it requires generating a full grid of data points with all possible combinations of values for those input variables that are tested for interaction. To overcome the problem of extrapolations over unpopulated regions of the input space, as well as problems caused by correlations, (Hooker, 2007) suggests imposing low weights for points from low-density regions. Unfortunately, this requires the use of external density estimation techniques and further increases complexity of the method.

We take a model comparison approach to interaction detection. In doing so, we do not need to calculate partial dependence functions to estimate predictor effects and we avoid the associated problem of spurious interactions from sparse regions. We believe this is a more direct approach to interaction detection.

9. Discussion

We presented a novel technique for detecting statistical interactions in complex data sets. The main idea is to compare the predictive performance of unrestricted models to restricted models, which do not contain the to-be-tested interaction. Although this idea is quite intuitive, there are significant practical challenges and few algorithms will work in this framework. We demonstrated that layered Additive Groves can be used in this approach due to its high predictive performance for both restricted and unrestricted models. Results on synthetic and real data indicate that we can reliably identify interactions.

Acknowledgements. We would like to thank Wes Hochachka, Giles Hooker, Steve Kelling and Art Munson for insightful discussions. This work was supported by grants ITR EF-0427914, SEI IIS-0612031, IIS-0748626 and NSF-0412930. Daria Sorokina was supported by a fellowship from Leon Levy foundation.

References

- Camacho, R. (1998). Inducing models of human control skills. *Proc. ECML'98*.
- Caruana, R., Elhawary, M., Fink, D., Hochachka, W. M., Kelling, S., Munson, A., Riedewald, M., & Sorokina, D. (2006). Mining citizen science data to predict prevalence of wild bird species. *KDD'06*.
- Christensen, R. (1996). *Plane answers to complex questions, the theory of linear models*. Springer.
- Friedman, J. (2005). RuleFit with R. <http://www-stat.stanford.edu/~jhf/R-RuleFit.html>.
- Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of Statistics*, 29, 1189–1232.
- Friedman, J. H., & Popescu, B. E. (2005). *Predictive learning via rule ensembles* (Technical Report). Stanford University.
- Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. *JMLR*, 3.
- Hooker, G. (2004). Discovering ANOVA structure in black box functions. *Proc. ACM SIGKDD*.
- Hooker, G. (2007). Generalized functional ANOVA diagnostics for high dimensional functions of dependent variables. *JCGS*.
- Jakulin, A., & Bratko, I. (2004). Testing the significance of attribute interactions. *Proc. ICML'04*.
- Pace, R. K., & Barry, R. (1997). Sparse spatial autoregressions. *Statistics and Probability Letters*, 33.
- Rasmussen, C. E., Neal, R. M., Hinton, G., van Camp, D., Revow, M., Ghahramani, Z., Kustra, R., & Tibshirani, R. (2003). Delve. University of Toronto. <http://www.cs.toronto.edu/~delve>.
- Ruppert, D., Wand, M. P., & Carroll, R. J. (2003). *Semiparametric regression*. Cambridge.
- Sorokina, D., Caruana, R., & Riedewald, M. (2007). Additive Groves of regression trees. *Proc. ECML*.
- Torgo, L. (2007). Regression DataSets. www.liacc.up.pt/~ltorgo/Regression/DataSets.html.