Evolving Probabilistically Significant Epistatic Classification Rules for Heterogeneous Big Datasets

John P Hanley  
University of Vermont  
43 Colchester Ave  
206 Perkins Hall  
1-802-922-5175  
jhanley@uvm.edu

Margaret J Eppstein  
University of Vermont  
33 Colchester Ave  
327 Votey Hall  
1-802-656-1918  
maggie.eppstein@uvm.edu

Jeffrey S Buzas  
University of Vermont  
16 Colchester Ave  
201 Henry Lord House  
1-802-656-2971  
jeff.buzas@uvm.edu

Donna M Rizzo  
University of Vermont  
33 Colchester Ave  
213 Votey Hall  
1-802-656-1485  
drizzo@uvm.edu

ABSTRACT

We develop an algorithm to evolve sets of probabilistically significant multivariate feature interactions, with co-evolved feature ranges, for classification in large, complex datasets. The datasets may include nominal, ordinal, and/or continuous features, missing data, imbalanced classes, and other complexities. Our age-layered evolutionary algorithm generates conjunctive clauses to model multivariate interactions in datasets that are too large to be analyzed using traditional methods such as logistic regression. Using a novel hypergeometric probability mass function for fitness evaluation, the algorithm automatically archives conjunctive clauses that are probabilistically significant at a given threshold, thus identifying strong complex multivariate interactions. The method is validated on two synthetic epistatic datasets and applied to a complex real-world survey dataset aimed at determining the drivers of household infestation for an insect that transmits Chagas disease. We identify a set of 178,719 predictive feature interactions that are associated with household infestation, thus dramatically reducing the size of the search space for future analysis.

Keywords

Hypergeometric PMF; epistasis; multivariate analysis; Chagas

1. INTRODUCTION

This work was motivated by the desire to mine data from a large socioeconomic survey aimed at identifying the drivers of household infestation with an insect that transmits Chagas disease, which if untreated is life-threatening. The disease is transmitted by insects in the subfamily Triatomine [3]. Approximately 110 million people live where the disease is endemic and of those, approximately 8 million are infected [10]. To decrease the risk of disease transmission, mitigation strategies, known as Ecohealth interventions, have been implemented to lessen the chance of household infestation of the Triatomine vectors by removing known hiding locations [15]. However, many areas where the disease is endemic have limited resources for these preventative household improvements. Detailed entomologic and socioeconomic surveys are conducted to try to identify the drivers of infestation [1] in order to determine the most cost-effective strategies to reduce infestation. Mining these complex Chagas survey datasets for useful information has proven to be a major challenge, due to a variety of factors, including missing data, imbalanced class outcomes, heterogeneity of drivers of infestation, non-independence of some features, and complex, possibly high-order, nonlinear interactions between many of the potential predictive features. Traditional statistical methods for survey analysis, such as logistic regression, are not suitable for analysis of the Chagas survey data because p-values are unreliable in large datasets [13] such as this.

The goal of this work is the development of a method to identify probabilistically significant epistatic interactions among input features correlated with a desired categorical outcome, where the input features may be comprised of multiple data types (e.g., nominal, ordinal, and continuous). For example, when the insect infests a house, it needs a food source and habitat (place to hide) at minimum. In addition, researchers believe that other factors (e.g., lights, windows) impact the attraction of the insect to the house. So in addition to habitat and food sources, the survey adds risk factors that suggest more interesting epistatic interactions of a higher order.

As a first step in the analysis of the Chagas survey dataset, our methodology is intended to deal with the massive number of complex multivariate interactions present in big datasets. A number of studies assume that individual features have main effects and use univariate logistic regression as a feature selection tool to reduce the size of the feature set that is to be mined for higher order interactions [4, 9]. Decision trees similarly enforce a hierarchical structure [17] based on the presence of main effects. Using main effects as a filter, however, precludes discovery of purely epistatic interactions that may drive a system, which may be the case for complex diseases [21], and our Chagas survey dataset.
We first tried applying ExSTraCS v1.0, a Michigan-style Learning Classifier System (LCS) specifically designed to search for heterogeneous epistatic interactions [19], to our dataset. To our knowledge, ExSTraCS is the best available method for analyzing large datasets when output categories are associated with more than one heterogeneous multivariate feature combination. However, we discovered that ExSTraCS was not well-suited to our Chagas survey dataset. Samples with too much missing data were problematic and needed to be removed, therefore causing us to throw away portions of our survey data. Also, our unbalanced dataset seemed to negatively impact results. In addition, ExSTraCS, as in most Michigan-style learning classifier systems [2], learns incrementally from individual samples rather than applying batch learning, thus requiring computationally expensive post-processing permutation tests to evaluate the significance of the identified features [20]. Furthermore, in our Chagas survey data, there may be ranges or sets of values for each input feature (i.e., predictor variable) that are associated with the same output category. Co-evolving feature ranges is not explicitly supported by ExSTraCS, but has been used in a few other genetics-based, machine-learning algorithms. For example, [14] and [17] supported co-evolution of ranges of feature values for continuous and ordinal features in conjunctive clauses (i.e., features with a range joined by ANDs), and [5, 6] supported coevolution of feature set values for nominal features in conjunctive clauses. Determining the statistical significance of evolved interaction terms is challenging in large datasets, and has been approached in various ways. As mentioned above, one option is to run post-processing permutation tests [20]. Others have tried to build significance tests directly into the fitness function. In [14], the authors propose two fitness functions for evaluating the conjunctive clauses. The first is the product of the accuracy and the coverage of the conjunctive clause; but this does not provide a natural threshold to identify important conjunctive clauses. The second, referred to as a CBayesian function, is the product of probabilities that, given the target output state, each evolved predictor variable has the evolved range. While this allows one to set a threshold for significance, it is based on a product of probabilities of univariate associations with the target class rather than on the probability of the multivariate association. Others [17, 18] use a statistics based fitness function using the z-score of the standard normal distribution that takes the following inputs: (1) coverage of the conjunctive clause, (2) proportion of observations with the target outcome, and (3) the number of observations matching both the conjunctive clause and the target outcome divided by the number of observations that match the conjunctive clause. The benefit of using a standard normal distribution is that p-value thresholds (e.g., p < 0.05) may be applied to the fitness values to identify statistically significant conjunctive clauses. However, p-values decrease inversely with the size of the dataset, making the p-value an unreliable statistic for “Big Data” applications [13].

To overcome this challenge, we propose using the hypergeometric probability mass function (PMF) [12] as a fitness function. The hypergeometric PMF may be thought of as a pseudo-Bayesian equation. One benefit of using the probability derived from the hypergeometric PMF is that it takes into account the size of the dataset and the distribution of the output categories. Therefore, unlike in the method of [17, 18], the ratio of output categories does not need to be balanced. Also, since the fitness is a probability, the user can choose a threshold that is easily interpreted and can compare probabilities below the threshold.

In Section 2, we describe an evolutionary algorithm (EA) to efficiently identify sets of probabilistically significant conjunctive clauses that represent multivariate interactions in large datasets. We first define the hypergeometric PMF fitness function, and then discuss how we incorporated this fitness function into an evolutionary algorithm using age-layers and an archive. In Section 3 we describe two synthetic epistatic datasets used for validation as well as the complex real-world survey dataset aimed at finding drivers of household infestation for an insect that transmits Chagas disease. We show results of these experiments in Section 4, and present concluding remarks in Section 5.

The primary intended contribution of this paper is a method to find collections of conjunctions with significant associations to a target class, thereby dramatically reducing the size of the search space for subsequent analysis in Big Data classification problems. How one chooses to subsequently analyze that set will depend on the nature and complexity of the dataset and the stakeholder’s objectives. For the relatively simple synthetic validation datasets, we illustrate how our genetic algorithm identifies a minimal disjunctive combination of these archived conjunctive clauses (i.e., conjunctive clauses joined by ORs), which simultaneously maximize coverage and accuracy. However, secondary mining of the large set of significant conjunctive clauses identified in the complex real world Chagas dataset will require more analyses, a topic we leave for future research.

2. ALGORITHMIC METHODS

2.1 Fitness Function

We seek sets of conjunctive clauses that have a probabilistically significant association with a certain class outcome k (e.g., infestation or disease). We define the fitness of a given conjunctive clause as follows:

\[ \text{Fitness(conjunctive clause)} = \left( \frac{X_{\text{tot}}}{N_{\text{tot}}} \right) - \frac{X_{\text{match}}}{n_{\text{match}}} \]  

(1)

where, \( N_{\text{tot}} \) is the total number of samples in the dataset, \( X_{\text{tot}} \) is the total number of those samples in the target class k, \( n_{\text{match}} \) is the number of samples whose features match a given clause, and \( X_{\text{match}} \) is the number of observations that match the clause and are in target class k. Eq. (1) is a hypergeometric PMF [12] and quantifies the likelihood that the observed association between the clause and the target class is due to chance; thus, lower values of this fitness function are indicative of potential association between a clause and a target class. We develop an evolutionary algorithm (described in the Section 2.4) to find clauses with low fitness (per Eq. 1), and archive all clauses whose fitness is less than a specified probability threshold \( T \). In the studies reported here, we used \( T = 10^{-6} \), which translates to a probability of 1 in a billion that a clause with fitness = \( T \) is randomly associated with the target class k.

2.2 Evolving Conjunctive Clauses

To support the co-evolution of feature sets and feature ranges, we represent multivariate interactions with conjunctive clauses of the following form:

\[ \text{PV}_i \in a_i \land \text{PV}_j \in a_j \land ... \land \text{PV}_k \in a_k \]  

(2)

where each \( \text{PV}_i \) represents a predictor variable i that may be nominal, ordinal, or continuous, and whose value lies in \( a_i \), a
specified range or set of values. The population is stored using a combination of a binary matrix (indicating feature presence/absence) and a parallel cell array (specifying the range or set of values for each feature present) (Figure 1). In this work, we allow the number of features present in a given clause to range from 1 to \( L \) (the number of features); we take this conservative approach in order to avoid a priori assumptions on the maximum order of interactions present in a given complex system.

The evolutionary algorithm is summarized in the pseudo code shown in Algorithm 1, and is described below. Line 1 of Algorithm 1 states that the evolutionary algorithm is run separately for each desired target outcome class \( k \), out of up to \( K \) possible classes. Lines 2 – 9 describe our implementation of the Age-Layer Population Structure (ALPS) created by [11]. We add a novel population (\( \text{Pop}_k \)) every 10 generations (\( \text{Gen}_k \)). In total, our algorithm has 10 “Age-Layers,” however, layers 1 – 9 follow the structure described in [11]; where the maximum age for layers 1, 2, \ldots, 8 is 10, 20,\ldots, 80, respectively, and the ninth layer has no age limit. Our tenth layer is used to archive the population of conjunctive clauses that have a proportion \( (x_{\text{max}}/N_{\text{tot}}) \) of the current output class that is greater or equal to the global proportion \( \left( X_{\text{tot}}/N_{\text{tot}} \right) \). Where the maximum age for layers 1, 2, \ldots, 8 is 10, 20,\ldots, 80, respectively, and the ninth layer has no age limit. Our tenth layer is used to archive the population of conjunctive clauses that have a proportion \( (x_{\text{max}}/N_{\text{tot}}) \) of the current output class that is greater or equal to the global proportion \( \left( X_{\text{tot}}/N_{\text{tot}} \right) \). In this work, we consider the order of features in finding associations to a given class \( k \).

Each generation, duplicate conjunctive clauses are removed from both the archived and non-archived populations (Algorithm 1, line 10 – 16 of Algorithm 1, and is described below). Lines 2 – 9 describe our implementation of the Age-Layer Population Structure (ALPS) created by [11]. We add a novel population (\( \text{Pop}_k \)) every 10 generations (\( \text{Gen}_k \)). In total, our algorithm has 10 “Age-Layers,” however, layers 1 – 9 follow the structure described in [11]; where the maximum age for layers 1, 2, \ldots, 8 is 10, 20,\ldots, 80, respectively, and the ninth layer has no age limit. Our tenth layer is used to archive the population of conjunctive clauses that have a proportion \( (x_{\text{max}}/N_{\text{tot}}) \) of the current output class that is greater or equal to the global proportion \( \left( X_{\text{tot}}/N_{\text{tot}} \right) \). This biases the algorithm toward conjunctive clauses that are useful in finding associations to a given class \( k \).

Each generation, duplicate conjunctive clauses are removed from both the archived and non-archived populations (Algorithm 1, line 10 – 16 of Algorithm 1, and is described below). Lines 2 – 9 describe our implementation of the Age-Layer Population Structure (ALPS) created by [11]. We add a novel population (\( \text{Pop}_k \)) every 10 generations (\( \text{Gen}_k \)). In total, our algorithm has 10 “Age-Layers,” however, layers 1 – 9 follow the structure described in [11]; where the maximum age for layers 1, 2, \ldots, 8 is 10, 20,\ldots, 80, respectively, and the ninth layer has no age limit. Our tenth layer is used to archive the population of conjunctive clauses that have a proportion \( (x_{\text{max}}/N_{\text{tot}}) \) of the current output class that is greater or equal to the global proportion \( \left( X_{\text{tot}}/N_{\text{tot}} \right) \). This biases the algorithm toward conjunctive clauses that are useful in finding associations to a given class \( k \).

The evolutionary algorithm is summarized in the pseudo code shown in Algorithm 1, and is described below. Line 1 of Algorithm 1 states that the evolutionary algorithm is run separately for each desired target outcome class \( k \), out of up to \( K \) possible classes. Lines 2 – 9 describe our implementation of the Age-Layer Population Structure (ALPS) created by [11]. We add a novel population (\( \text{Pop}_k \)) every 10 generations (\( \text{Gen}_k \)). In total, our algorithm has 10 “Age-Layers,” however, layers 1 – 9 follow the structure described in [11]; where the maximum age for layers 1, 2, \ldots, 8 is 10, 20,\ldots, 80, respectively, and the ninth layer has no age limit. Our tenth layer is used to archive the population of conjunctive clauses that have a proportion \( (x_{\text{max}}/N_{\text{tot}}) \) of the current output class that is greater or equal to the global proportion \( \left( X_{\text{tot}}/N_{\text{tot}} \right) \). This biases the algorithm toward conjunctive clauses that are useful in finding associations to a given class \( k \).

The only other time conjunctive clauses are removed from the population is if the maximum size of the non-archived population (\( \text{Pop}_k \)) is exceeded. If this occurs, then for each non-archived age layer, only the two most-fit conjunctive clauses for each feature variable are saved for the next generation. These two most-fit rules are selected one feature at a time to maintain diversity in the population. When selecting most-fit rules without accounting for feature representation, we observed that some more complex features (e.g., larger ranges) and interactions did not evolve to the desired signal. Repeat conjunctive clauses are prevented by selection without replacement. To avoid bias, the order of features considered is randomized in this process.

Algorithm 2, describes the creation of novel populations in the first layer (\( \text{Pop}_k \)). Line 2 states that an observation \( n' \) is randomly selected from the observations equal to the target outcome \( k \). All random numbers were generated from a uniform distribution. Each conjunctive clause is created as follows. We randomly select the number of features to be present in the clause (Algorithm 2, Line 3), and then randomly select the features for the clause. If the selected feature is continuous or ordinal, the upper and lower bounds of the conjunctive clause for the feature variable are set to the corresponding value in the randomly selected observation \( n' \) (Algorithm 2, Lines 6 – 7). For nominal features, the value is set to the corresponding value of \( n' \). At no point can the conjunctive clause contain blank ranges or sets for selected features.

Algorithm 3 describes mutation and crossover for the evolutionary algorithm. In each generation, all the conjunctive clauses in \( \text{Pop}_{kA} \) undergo either mutation or crossover. However, for the archive (\( \text{Pop}_{kA} \)) only up to two randomly selected conjunctive clauses for

![Figure 1](image-url)
the current age-layer or the adjacent younger age-layer, if one
selection with replacement and a tournament size of 3, from either
crossover, a second parent is randomly selected, using tournament
result of there being no limit on the population size of
each other datasets as well. If a conjunctive clause is selected to undergo
all of the “true” bivariate interactions, so we used this rate for the
probability of crossover
testing the algorithm on the [21] dataset, we found that the
probability
exists. The parents then undergo uniform crossover and produce
one offspring (Algorithm 3, Lines 3 – 5).
If a conjunctive clause is selected to undergo mutation, then the
probability that a given feature will undergo mutation is
a standard rule of thumb [8], although we force at least one feature
to be selected (Algorithm 3, Line 7). If the selected feature value is
1 in the binary matrix (Fig 1A), then there is a 1/
locus changed to 0
if locus is still 1
if locus is ordinal/continuous
Change one bound
else
Remove, change, or
delete 1 category
elseif locus is 0 change to 1
if locus is ordinal or continuous
Randomly select a range
else
Randomly select a category
end
end

Algorithm 2: Initialize Novel Populations
1
2
3
4
5
6
7
8
9
each PV undergo mutation or crossover each generation. This is a
result of there being no limit on the population size of PopL. When
testing the algorithm on the [21] dataset, we found that the
probability of crossover PrC = 0.25 consistently enabled us to find
all of the “true” bivariate interactions, so we used this rate for the
other datasets as well. If a conjunctive clause is selected to undergo
crossover, a second parent is randomly selected, using tournament
selection with replacement and a tournament size of 3, from either
the current age-layer or the adjacent younger age-layer, if one

Algorithm 1: Evolutionary Algorithm (EA)
1 for k = 1 to K
2 | for gen = 1 to Gen
3 | if gen = 1
4 | Run Algorithm 2 – Create PopN
5 | elseif mod(gen, GenN) is 1
6 | Run Algorithm 2 – Create PopN
7 | Run Algorithm 3 – Evolve PopNA & PopA
8 | else
9 | Run Algorithm 3 – Evolve PopNA & PopA
10 | for i = 1 to Total Offspring
11 | Calculate Fitness
12 | if Fitness < T & xmatch / nmatch ≥ Xtot/Ntot
13 | Place offspring in PopA
14 | elseif xmatch/nmatch ≥ Xtot/Ntot
15 | Place offspring in PopNA
16 | end
17 | Remove duplicates from PopNA & PopA
18 | if PopNA > ALNA x L
19 | for j = 1 to ALNA
20 | for each PV (in random order)
21 | Save 2 most-fit clauses
22 | end
23 | end
24 | end
25 | end

Algorithm 3: Introduce Genetic Variation
1 for i = 1 to Pop
2 | Randomly (PrX) select crossover or mutation
3 | if crossover
4 | Select mate from age layer or layer below
5 | Crossover is uniform with 1 offspring
6 | elseif mutation
7 | Select loci p = 1/L (min = 1 locus)
8 | for j = 1 to total loci
9 | if locus is 1 in binary matrix (Fig 1A)
10 | p =1/L locus changed to 0
11 | if locus is still 1
12 | if locus is ordinal/continuous
13 | Change one bound
14 | else
15 | Remove, change, or
delete 1 category
16 | elseif locus is 0 change to 1
17 | if locus is ordinal or continuous
18 | Randomly select a range
19 | else
20 | Randomly select a category
21 | end
22 | end
23 | end
newly added feature will create a fit offspring (and thus be retained) is small.

The values for control parameters of this study (Table 1) were selected as a combination of published recommendations in the literature and parameters that were empirically determined as the most effective at quickly identifying the 4 true conjunctions in the first validation dataset (such as the probability of crossover \( Pr_c \)), described earlier.

Table 1. EA parameters used of evolving conjunctions.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total # of Predictor Variables ( (L) )</td>
<td>Dataset dependent</td>
</tr>
<tr>
<td>Threshold ( (T) )</td>
<td>( 10^9 )</td>
</tr>
<tr>
<td># Non-Archived Layers ( (AL_{NA}) )</td>
<td>9</td>
</tr>
<tr>
<td>Novel Population ( (Pop_N) )</td>
<td>( 2 \times L )</td>
</tr>
<tr>
<td>Non-Archive Pop. ( (Pop_{NA}) )</td>
<td>Max = ( AL_{NA} \times L )</td>
</tr>
<tr>
<td>Archive Population ( (Pop_A) )</td>
<td>Max = ( \infty )</td>
</tr>
<tr>
<td>Generations ( (Gen) )</td>
<td>Dataset dependent</td>
</tr>
<tr>
<td>Generations Until Novel Pop ( (Gen_{nv}) )</td>
<td>10</td>
</tr>
<tr>
<td>Crossover Function ( (Fx) )</td>
<td>Uniform</td>
</tr>
<tr>
<td>Probability of Crossover ( (Prx) )</td>
<td>0.25</td>
</tr>
<tr>
<td>Mutation Function ( (Fm) )</td>
<td>{uniform, ( p_m = 1/L }}</td>
</tr>
<tr>
<td>Crossover Mate Selection ( (Off) )</td>
<td>{tournament, size = 3}</td>
</tr>
<tr>
<td># Archive Offspring ( (Off_A) )</td>
<td>( 2 \times L )</td>
</tr>
</tbody>
</table>

2.3 Evolving DNF Rules

After sets of probabilistically significant conjunction clauses have been identified using Algorithm 1, subsequent mining of this reduced search space can reveal useful combinations of these conjunctions. For example, by treating each conjunction in the archive as a binary feature, one can use a genetic algorithm (GA) to identify disjunctive combinations of these features (i.e., clauses in disjunctive normal form, or DNF). DNFs are thus represented as bit strings, where bit values represent the presence or absence of each conjunctive clause from the archived dataset. For this step, we defined fitness of a clause as follows:

\[
\text{DNF}_{.} \text{Fitness} (\text{clause}) = \overline{Acc} \times \overline{Cov} \times \overline{NonRep} \times \overline{TotCov} \tag{3}
\]

Where \( \overline{Acc} \) is the weighted average accuracy \( (\Sigma_{\text{match}}/\Sigma_{\text{total}}) \) of the conjunctive clauses, \( \overline{Cov} \) is the average coverage \( (x_{\text{match}}/x_{\text{total}}) \) of the conjunctive clauses, \( \overline{NonRep} \) is the number of samples with output \( k \) that conjunctive clauses in the DNF collectively cover divided by the sum of the number of samples with output \( k \) each conjunctive clause individually covers, and \( \overline{TotCov} \) is the number of samples with output \( k \) that conjunctive clauses collectively cover divided by the total number of observations in the dataset with output \( k \). The GA was trying to maximize the fitness defined in Eq. (3).

The GA is based on the Age-Fitness Pareto Optimization developed by [16], with pseudo code shown in Algorithm 4 and control parameters in Table 2. In the first generation, a population of 20 DNFs is generated by randomly selecting between 1 – 20 conjunctive clauses to be present (Algorithm 4, Line 3). Each DNF is evaluated using the fitness function Eq. (3); and each DNF is assigned an age = 1 (Algorithm 4 Lines 4 – 5). For subsequent generations, each DNF undergoes either mutation or crossover with \( Pr_c \) (Algorithm 4, Line 7). DNFs selected for mutation will have a \( p_m \) of having each bit flipped consequently adding and removing conjunctive clauses. All offspring derived from mutation are given the same age as the parent. DNFs selected for crossover undergo either set union or set intersection (\( Pr_u \) [7]). The second parent is randomly selected, using tournament selection with replacement and a tournament size of 3. If two parents do not have a single conjunctive clause in common then crossover is forced to be union. All offspring are given the age of the oldest parent. Offspring DNFs are then evaluated using the fitness function (Eq. 3) (Algorithm 4, Line 8) and the age of both offspring and parents will increase by one (Algorithm 4, Line 9). The offspring and parents are plotted based on their age and fitness and only those DNFs that lie along the non-dominated front, where dominance is determined by highest fitness and lowest age, are archived for the next generation (Algorithm 4, Line 10). A new population of 20 DNFs is generated by randomly selecting 1 – 20 conjunctive clauses (Algorithm 4, Line 11). The fitness from Eq. (3) of the new DNFs is evaluated.
and all new DNFs are assigned an age of 1 (Algorithm 4, Lines 12 – 13). The non-dominated DNFs are combined with the new DNFs and evolved in the subsequent generation.

3. Test Datasets and Experiments
We applied the evolutionary algorithm to two synthetic datasets for validation, and for preliminary analysis of the Chagas dataset.

3.1.1 Synthetic Dataset 1
The first validation dataset was designed by [21] to represent a synthetic Single Nucleotide Polymorphism (SNP) gene association dataset for a complex disease that incorporates both genetic heterogeneity and epistasis. This balanced dataset contains 1,600 samples each with 20 input feature variables, with exactly half of the samples associated with the “disease”. Each input feature represents a SNP and is a ternary representation of whether the SNP is homozygous major, heterozygous, or homozygous minor. The synthetic dataset was designed such that no features had significant main effects and that four bivariate feature interactions (between two sets of two features) had strong correlations to different subsets of “diseased” observations, as shown in Table 3.

Table 3. The 4 “true” 2-way interactions in the [21] dataset (synthetic dataset 1). The last three columns contain the accuracy, coverage, and fitness of the rules.

<table>
<thead>
<tr>
<th>PV₀</th>
<th>PV₁</th>
<th>PV₂</th>
<th>PV₃</th>
<th>Acc.</th>
<th>Cov.</th>
<th>Fit.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>–</td>
<td>72%</td>
<td>27%</td>
<td>1.1x10⁻¹²</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0</td>
<td>–</td>
<td>74%</td>
<td>23%</td>
<td>5.7x10⁻¹²</td>
</tr>
<tr>
<td>3</td>
<td>–</td>
<td>–</td>
<td>0</td>
<td>66%</td>
<td>28%</td>
<td>4.2x10⁻¹²</td>
</tr>
<tr>
<td>4</td>
<td>–</td>
<td>–</td>
<td>1</td>
<td>71%</td>
<td>21%</td>
<td>8.7x10⁻¹³</td>
</tr>
</tbody>
</table>

For this dataset, we ran 50 repetitions of the evolutionary algorithm (Algorithm 1) for 1,000 generations to try to find associations with the “diseased” class. We recorded the number of generations and fitness evaluations for each generation as well as the generation in which all the known a priori interactions were found. Afterwards, the archived conjunctive clauses were run through the GA (Algorithm 4) to find the “best” disjunctive normal form (DNF) of the archived rules. The GA was run 50 times; for each run we recorded the total number of generations and fitness evaluations to find the four “best” DNF rules in (Table 3) in the dataset associated with “diseased” observations.

3.1.2 Synthetic Dataset 2
While synthetic dataset 1 tested the algorithm’s ability to identify the four correct bivariate epistatic interactions in this balanced dataset, it did not enable testing of the performance in the face of other complexities present in our Chagas dataset. We thus designed a second synthetic dataset to have 1) nominal and discrete input features, 2) higher-order epistatic interactions, 3) missing data, 4) heterogeneous associations with the same output class, and 5) imbalanced class outcomes. Specifically, the dataset consists of 25 input features, of which 17 are nominal (with values ranging from 1 to 3, 4, 6, or 9) and 8 are discrete integers (with values ranging from 1 to 15 or 20). There are 1,000 simulated samples, of which 339 have output class = 1, representing “diseased,” and the remaining 661 have output class = 0, representing “not diseased.” Of the “diseased” samples, 144 were designed with a 4-way interaction (between 2 nominal and 2 discrete features) that is strongly correlated with disease, and a separate set of 195 samples have a 3-way interaction (between 3 nominal features) that is strongly correlated with the disease, as shown in Table 4. After randomly removing 20% of the data (to simulate missing data), there remained 58 4-way and 89 3-way intact interactions associated with class 1. Both “true” conjunctions had 100% accuracy but different levels of coverage and fitness (Table 4). The remaining 16 features contained randomly selected values.

Table 4. The 2 “true” 3-way and 4-way interactions built into synthetic dataset 2. The Ns and Ds below each feature stand for nominal and discrete feature types, respectively. Also included is the range of each of the features in the true interactions. The last three columns contain the accuracy, coverage, and fitness of the rules.

<table>
<thead>
<tr>
<th>PV₀</th>
<th>PV₁</th>
<th>PV₂</th>
<th>PV₃</th>
<th>PV₄</th>
<th>PV₅</th>
<th>PV₆</th>
<th>Acc.</th>
<th>Cov.</th>
<th>Fit.</th>
</tr>
</thead>
<tbody>
<tr>
<td>{1 – 15}</td>
<td>{1 – 6}</td>
<td>{1 – 20}</td>
<td>{1 – 6}</td>
<td>{1 – 9}</td>
<td></td>
<td></td>
<td>100%</td>
<td>17%</td>
<td>1.7x10⁻²⁹</td>
</tr>
<tr>
<td>{5 – 8}</td>
<td>{2, 4}</td>
<td>{1}</td>
<td>{1 – 2}</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>100%</td>
<td>26%</td>
<td>2.6x10⁻⁴⁶</td>
</tr>
</tbody>
</table>

For synthetic dataset 2, we ran 25 repetitions of the evolutionary algorithm (Algorithm 1). The algorithm was stopped when both the known a priori multivariate interactions were discovered and the number of generations and fitness evaluations were recorded. Again, the archived conjunctive clauses from the first iteration of the evolutionary algorithm were run through the GA (Algorithm 4) to find the best DNF.

3.1.3 Real-World Chagas Dataset
From August to October 2011, teams comprised of personnel from the Escuela de Biologia, La Universidad de San Carlos Guatemala and the Guatemalan Ministry of Health Office of Vector-Borne Diseases conducted entomological and socioeconomic surveys of 1,132 houses in five villages in Olopa, Chiquimula, Guatemala. Informed consent was obtained from all human adult participants and from parents or legal guardians of minors. This project received ethical clearance from the Ministry of Health in Guatemala, La Universidad de San Carlos bioethics committee, and the Panamerican Health Organization. This Chagas dataset is not publicly available since it is human subject research data that is not possible to anonymize. The surveys contain 61 risk factors that experts believe are associated with infestation of households with Triatoma dimidiata, a vector of Chagas disease. Thirteen of the risk factors are ordinal/continuous and the remaining 48 are nominal. The total number of houses that are infested is 314 (28%). Sixteen percent of the data are not present and are considered missing (rather than imputed). For the Chagas infestation dataset, the evolutionary algorithm (Algorithm 1) was run once for 1,500 generations to seek associations with households that are infested. Further analysis of the archived significant conjunctions for this dataset is beyond the scope of this paper and remains a subject for future study.

4. RESULTS
4.1 Simulated Data Results
An exhaustive search of the [21] dataset (for up to 20-way interactions) found over 10⁹ possible feature combinations present in the observations. Of the ~10⁹ feature combinations, only 1,270 are correlated with having the disease and have a fitness score (by Eq. 1) less than the user defined threshold of 10⁻⁶; three of these were of order nine.
Algorithm 1 found over 90% of the 1,270 conjunctive clauses that have a strong correlation with each output category in under 1,000 generations, in all 50 repetitions (Table 5). All four of the “true” bivariate feature combinations designed to correlate with the disease (from Table 1) were identified and archived in under 143,000 fitness evaluations; note this is significantly less than an exhaustive search of all $10^9$ possible conjunctive clauses. There are over $1.4 \times 10^6$ possible 9-way interactions in the dataset; yet in each run, Algorithm 1 was able to find at least one of the 3 significant 9-way interactions (fitness < $10^{-9}$) in less than $1.4 \times 10^5$ evaluations. Thus the algorithm can find high order signals quickly in large search spaces (Table 5).

4.2 Chagas Data Results

The evolutionary algorithm archived 178,719 conjunctive clauses associated with households infested with the Chagas vector (up to 5-way interactions are shown in Figure 2). Only 11 of the 61 features are present in archived univariate conjunctive clauses. However, 46, 55, 57, 60 and 61 features are present in archived 2-, 3-, 4-, 5-, and 6-way interactions, respectively. Not surprisingly, the lower-order interactions and main effects tend to have higher coverage and lower accuracy, while high-order interactions tend to have higher accuracy and lower coverage (Figure 2). Figure 2 shows that 2-way interactions have higher accuracy and coverage than main effects at every point in the range except at the very highest levels of coverage. Also 3-, 4-, and 5-way interactions have higher accuracy and coverage than main effects at every point below 90% coverage. The plurality of the conjunctive clauses have 8-way interactions and the evolutionary algorithm was even able to find one 22-way interaction with 67% accuracy and 12% coverage. Note that one cannot infer dominance of the conjunctive clauses shown in Figure 2, since different clauses cover different parts of the space.

Figure 2. Coverage and accuracy of the 178,719 archived interactions associated with household infestation. Lower order conjunctive clauses are plotted on top of higher order.

5. CONCLUSIONS

We demonstrated a new methodology for efficiently searching for probabilistically significant multivariate interactions in complex heterogeneous “Big Datasets.” The algorithm was validated on two synthetic datasets, one that contained purely epistatic bivariate interactions and another that contained a 3-way and a 4-way
interaction, in addition to other complexities (nominal and discrete input features, 3- and 4-way interactions, missing data, heterogeneous associations with the same output class, and imbalanced class outcomes). Further experimentation has shown that lowering the probability threshold for the archive reduces the computational time for the evolutionary algorithm to find the true interactions in these synthetic datasets. However, in practice this comes with the risk of eliminating true signals for underrepresented segments of observations. We applied the method to a complex real-world Chagas infestation survey dataset and were able to archive many strong multivariate interactions. However, further analysis and interpretation of this set of archived conjunctions for the Chagas dataset is beyond the scope of this paper and remains a subject for future study.

6. ACKNOWLEDGEMENTS

We would like to thank researchers at La Universidad de San Carlos Guatemala, especially Carlota Monroy and Dulce Bustamante, for their help in collecting and managing the Chagas infestation dataset. We also thank the Guatemalan Ministry of Health for their help with collecting the data. This work was supported, in part, by Vermont EPSCoR with funds from the National Science Foundation Grant DBC-EID-1216193 (JPH) and EPS-1101317 (DMR).

7. REFERENCES


