TEXT and rectangles in blue will NOT show on printed copy

Type final title of thesis or dissertation (M.S. and Ph.D.) below . If your title has changed since your submitted an Application for Graduate Degree, notify Graduate Office.

Applying genetic algorithms to the problem of variable selection in large datasets with interaction terms

A Dissertation

Presented to

the faculty of the School of Engineering and Applied Science

University of Virginia

in partial fulfillment

of the requirements for the degree

Doctor of Philosophy

by

Name

Chee Chun Gan

Month degree is awarded May

Year 2016

APPROVAL SHEET

The dissertation

is submitted in partial fulfillment of the requirements

for the degree of

Doctor of Philosophy

Chee Chun Gan

AUTHOR *signature*

The dissertation has been read and approved by the examining committee:

Please insert committee member names below:

Gerard Learmonth

Advisor

Donald Brown

Laura Barnes

Abigail Flower

Douglas Lake

Accepted for the School of Engineering and Applied Science:

James H. Ay

Dean, School of Engineering and Applied Science

Month degree is awarded May

Year 2016

Print Form

Applying genetic algorithms to the problem of variable selection in large datasets with interaction terms

---

Author:

Chee Chun Gan

Advisor:

Dr. Gerard Learmonth

A thesis presented to

The Faculty of the School of Engineering and Applied Science

University of Virginia

In partial fulfillment of the requirements of the degree

Doctor of Philosophy in Systems and Information Engineering

May 2016

Advisory Committee members:

Dr. Donald Brown, Dr. Gerard Learmonth, Dr. Abigail Flowers, Dr. Laura Barnes, Dr. Douglas Lake

Abstract

Variable selection is a key step in the development of predictive models. When the size of the dataset is relatively small, greedy algorithms such as stepwise selection perform well in the selection of informative variables. However, as the size of the dataset increases, the challenges faced by such variable selection methods increases rapidly. The addition of interaction terms drastically increases the complexity of the variable selection problem, rendering greedy stepwise selection ineffective.

Past research on the topic has seldom included the effect of interaction terms on predictive modeling. Part of the reason may be the aforementioned difficulty involved in the variable selection process when considering a large dataset. Another possibility is the tradeoff between model accuracy and complexity, where the benefits from including interaction terms may be marginal. However, in certain applications such as medical diagnosis models, any marginal increase in predictive ability may lead to significant improvements in terms of lives saved. In addition, information obtained during the variable selection process such as which interaction terms are significant may serve as a guide for future research efforts to explore why such interaction terms exist among certain primary predictors.

A genetic algorithm (GA) is developed in this study to handle the expanded search space of primary and interaction terms for variable selection. While GAs have been used for variable selection in the past, the chromosome formulation and selection process must be modified to accommodate interaction terms in large datasets. The GA framework is highly flexible and is able to handle a large variety of different models simply by choosing the appropriate fitness function. Experimental runs show that there is benefit to including interaction terms in large datasets in addition to main effects.

Acknowledgements

First of all, I would like to thank my advisor, Professor Gerard Learmonth, for all the guidance, care and wisdom he has shown me through my years at the University of Virginia. I have learned so much under his tutelage and support, without which this project would never have gotten off the ground. I would also like to thank the faculty of the Department of Systems and Information Engineering, for the knowledge they have imparted to the students enrolled in the program. On a personal level, my wife, Siew Wei, as well as my son, Calvin, have been unwavering in their support and encouragement during my studies and were my biggest source of motivation.

Contents

Chapter 1 - Developing the Genetic Algorithm Framework

1.1 Introduction

Since time immemorial, people have sought to understand what affects the world around them. As people began to use predictive modelling to help understand complex processes, it became apparent that the choice of which variables to use in a model is a vital step. A modeler could choose to simply include every piece of information available. While such a model may perform well on the specific dataset it was built on, due to the variance-bias tradeoff it is likely to not be easily applicable to other datasets. On the other hand, if too few variables are chosen there could be valuable information left out of the model that could greatly increase performance.

Most models try to adhere to the principle of Occam's Razor: the simplest solution is often the correct one. Thus, modelers should try to design the simplest models (with the fewest predictors) that still perform adequately. As a consequence, much work has been done to develop variable selection methods that aim to provide information on which predictors should be included in a given model. However, despite being well-studied, the myriad complexities involved in modelling make it difficult to determine optimal variable selection methods. For example, different models can often have different performance measures, which will affect the efficacy of various variable selection techniques. As a result, most variable selection methods are heuristics which seek to evaluate candidate variables according to some statistical metric. Stepwise selection [1], which is one of the most commonly used variable selection methods, ranks candidate variables according to the marginal benefit of including said variable using Akaike's Information Criterion (AIC) [2], Bayesian Information Criterion (BIC) or Mallows C_p. [3]. Ensemble methods, such as random forests [4], have also become increasingly popular.

The recent focus on Data Science has generally been a boon to modelers by providing a wealth of data with which to develop models. Previously, one of the most common obstacles faced by modelers was the lack of reliable data to develop accurate models. However, we are now beginning to encounter problems where we face problems at the other end of the spectrum :

too much data is available and current variable selection methods are not well-designed to handle them.

An important note is that most of the common variable selection techniques are greedy algorithms. In general, such methods rank each potential variable according to the marginal benefit of including the variable in the current model (evaluated based on the chosen criterion), and choose to include the variable that provides the most benefit at that point. However, these methodologies are unable to account for situations where a variable provides a greater benefit when included together with one or more other specific variables. There are many real world cases where such "the whole is greater than the sum of its parts" phenomena can be observed. Models may account for such effects by the inclusion of "interaction terms", additional variables that indicate the joint effect of 2 or more primary terms. However, simply considering all possible 2-way interactions (joint terms for all possible pairs of variables) will cause the search space to increase in a combinatorial fashion. A model with 100 potential primary variables results in a potential search space of 5050 variables (100 + 4950 possible 2-way interactions). Therefore, we can see that modern Data Science problems can quickly cause scalability issues with most common variable selection methods, which now need to evaluate the marginal benefit of many more variables at each step in the process.

Compounding the problem is the fact that the detection of interaction terms is by itself also a complex problem with no standard methodology for definitely detecting significant interactions, causing modelers to be unable to easily pre-screen potential interaction terms. As an exhaustive search of all possible combinations of interaction terms is often not feasible even for a small number of variables, most modelers rely on expert knowledge or careful pre-processing and examination of the data in order to pick out potential interaction terms for inclusion in candidate models. In both cases, the efficacy decreases significantly as the number of variables increases. For models with hundreds of variables, it is possible that even domain experts may not be aware of potential interaction effects. Furthermore, one of the potential benefits of modeling with Data Science techniques is to gain knowledge about the problem space, such as discovering previously unforeseen interactions amongst predictor variables.

Relying on expert knowledge to predetermine “sensible” interaction terms would necessarily exclude such discoveries. It also becomes increasingly difficult to manually examine data for potential interaction terms as the number of variables grows. Common methods, such as examining data visualizations to spot correlations, quickly become very unwieldy and are also highly subjective.

It quickly becomes apparent that a better methodology for traversing the search space of potential predictor variables is required. Genetic algorithms (GAs) present a possible solution. GAs are a class of evolutionary algorithms that emulate the biological process of natural selection via “survival of the fittest”. There are numerous examples in the literature where GAs have been used for variable selection in various models, as one of the advantages of the GA approach is that it is independent of the actual model used. Herrero and Ortiz [5] examine the use of GAs for variable selection with partial least squares regression of various polarographic and stripping voltammetric data sets and conclude that GAs often provide improved predictive power, as well as some qualitative information on the problems considered. Gayou et al [6] applied a genetic algorithm for variable selection in a logistic regression model to predict radiotherapy treatment outcomes. However, there exists a gap in the literature pertaining to utilizing GAs together with interaction terms. One possible reason is the aforementioned desire to keep models as sparse as possible. The inclusion of interaction terms greatly increases the solution space, and depending on the data set may not necessarily yield significant improvements in predictive capability while greatly increasing the computational cost.

Recently, two factors have come into play that warrant taking a closer look at the potential benefits of exploring the interaction terms in the search space. Firstly, the increasing availability of large scale computing computer using parallelization and cloud computing means that it is relatively inexpensive to perform large scale (but not necessarily complex) calculations such as exploring the expanded set of potential predictors including interaction terms. Secondly, there is an increasing number of very large and complex datasets becoming available that may contain unforeseen, yet significant, interactions between variables. With a larger number of predictors comes a larger chance that some of those predictors may boost the predictive

capabilities of the model when included together as a subset. In certain fields, such as medicine, even a relatively small increase in model performance can mean the difference between life and death for some patients. These two factors considered in conjunction may warrant taking a closer look at improving predictive models by actively searching for interaction terms that may provide a slight increase in performance.

1.2 Genetic Algorithms

Genetic algorithms were first proposed by John Holland [7]. While not originally focused on optimization, it has become a common heuristic used for optimization purposes. As an evolutionary algorithm, GAs incorporate two key concepts from the biological process of natural selection: mutation and crossover. It uses these concepts to iteratively explore the solution space and exploit candidate solutions that perform well by pitting a number of candidate solutions (the population) against each other in successive generations.

The first step in setting up a GA is encoding potential solutions into a chromosome. A chromosome is a vector that contains information about the key parameters in a candidate solution. For example, a GA used for variable selection in a logistic regression model could have a chromosome represented by a string of binary values of length n, where n is the total number of possible variables. A value of 0 at string index i would indicate that the ith variable is not included, while conversely a value of 1 would indicate that the ith variable is included in the candidate solution. For example, Figure 1.1 below shows a sample chromosome for a model with 6 potential variables. The sample chromosome represents a model with the 2^{nd}, 3^{rd}, and 6^{th} variables included.

Figure 1.1 : Sample chromosome for main effects variable selection

0	1	1	0	0	1

After formulating the chromosome structure, a number of chromosomes are generated to form the initial population. The generation of the initial population can be performed using a variety

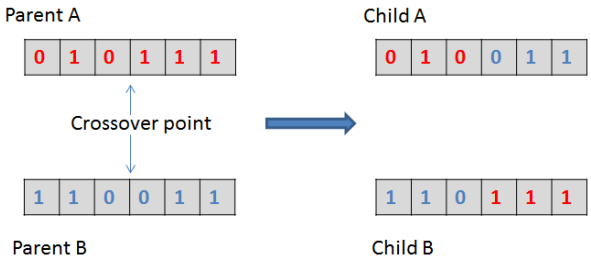
of methods, the most common being random generation by selecting each bit value in each chromosome according to a random distribution. The population can also be seeded with "good" solutions found through alternative methods in order to reduce the time spent exploring the solution space for viable solutions. Pre-seeding the population also weights the process more towards exploitation rather than exploration.

Once the initial population has been created, the algorithm proceeds to modify the individual chromosomes in succeeding generations via natural selection. In each generation, the performance of each member chromosome is evaluated using a fitness function. The fitness function is independent from the rest of the GA, hence making the GA a robust tool that can be used in a variety of models. For example, in a linear regression model the fitness function could be the adjusted R^2, the AIC, the BIC or even a weighted combination of the previous 3 measures.

After determining the fitness levels of all members of the population, a selection procedure is then used to choose several parent chromosomes. One common selection method is tournament selection, where candidates are chosen randomly to participate in a "tournament" during which the fitness values of competing chromosomes are compared, with the winner being selected as a parent chromosome. This parallels the biological process of natural selection where more fit individuals in a population have a greater chance of reproducing and passing on their genes to their offspring. Other selection methods include randomly selecting parent chromosomes with increasing probability corresponding to increasing fitness values, or simply ranking the candidate chromosomes and using the top performers as parents.

Once parent chromosomes have been selected, the crossover operation is used to generate offspring, or child chromosomes. Again, there are various forms of crossover operators used with the underlying notion of combining the genes from multiple (usually two) parent chromosomes into a single offspring. The most basic crossover operator is a fixed point crossover, with the crossover point usually being the midpoint of the chromosome. Figure 1.2 below shows a simple example of a fixed point crossover with two parent chromosomes A and B, with the crossover point being the chromosome midpoint.

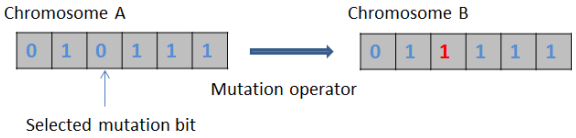
Figure 1.2 : Fixed point crossover



The underlying notion behind the crossover operator is that a high-performing parent chromosome should contain certain elements that contribute to its fitness score. In the case of a variable selection problem, it could be that high performing chromosomes contain a larger ratio of the "correct" variables. By combining the chromosomes of two parents, the crossover operator attempts to generate children which also have a high likelihood of equal or improved performance. This concept is related to the "building block hypothesis" [8] which postulates that over time, a GA will perform well by combining multiple short chromosome segments (which have high fitness) into longer, better performing chromosomes. The crossover operator can be applied according to a predefined probabilistic parameter setting. For example, a crossover probability of 0.5 would indicate that a pair of parents would have their chromosomes combined half the time. The other half of the time would see both parents being passed on to the next generation without mixing their chromosomes, similar to elitist selection.

The mutation operation is similar to neighborhood search or hill-climbing methods, where a small change is made to an existing candidate solution in order to explore solutions that are near the original solution in the search space. It is also necessary as a way to introduce novel solutions into the population, as otherwise after several generations the population would lose diversity by consisting only of various recombinations of the original population members. Similar to the crossover operator, the mutation operator is usually applied according to a predefined probabilistic parameter setting but is set to be much lower than that of crossover in order to avoid excessive disruption to the chromosome.

Figure 1.3 : Random mutation of single bit



The processes of selection, crossover and mutation taken together form the heart of most GAs. When viewed from the framework of exploration vs exploitation, crossover and mutation serve to explore the solution space in various degrees (crossover provides larger scale changes while mutation can adjust individual bits in the chromosome) while the selection process promotes exploitation of the best currently found solutions by using them as jump off points for exploration. The balance between exploration and exploitation must be adjusted for every application of the GA.

1.3 Interaction terms

The primary motivation in this study of using GAs in variable selection is to handle the inclusion of interaction terms in large scale datasets. For a regression model with a set of predictors X = $\{x_1,x_2,x_3\}$, a model with only main effects terms against the response Y is shown in Equation 1.1 below:

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 \qquad (1.1)$$

The abovementioned model allows for the effect of each predictor to be isolated and analysed. However, it is possible that the true model contains joint effects that cannot be isolated and can only be modeled using interaction terms. For first order interaction terms, this results in the following fully expanded model in Equation 1.2:

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_{1,2} + \beta_5 x_{1,3} + \beta_6 x_{2,3} \qquad (1.2)$$

In the expanded model, $x_{i,j}$ is the interaction term representing the joint effect of variables x_i and x_j. Interaction terms are symmetric, thus $x_{i,j}$ and $x_{j,i}$ are equivalent.

Thus far, interaction terms have been relatively neglected when it comes to Data Science modelling. Part of the reason could be that Data Science models inherently have to deal with many scalability problems that are greatly amplified by including interaction terms. With a small set of predictors, it is relatively simple to consider all possible interaction terms and eliminate non-significant terms with common methods such as stepwise selection or even exhaustive search. However, as the predictor set becomes larger the set of potential interaction terms grows combinatorially (for k-way interactions with n predictors, the number of interaction terms is $\binom{n}{k} = \frac{n!}{k!(n-k)!}$), quickly outstripping the ability of modelers to manually detect potential interactions using expert judgement or data visualization. The expanded set of variables can also cause problems for some common variable selection methods which try to evaluate the marginal benefit of including each variable, as a much larger number of variables have to be considered at each step. A second reason could be that interaction effects are often minor compared to the effects of primary predictors, thus including them is not cost effective in terms of the tradeoff between improved accuracy and model sparsity.

Despite the abovementioned complications, it is clear that there are some situations where it would still be worthwhile to search for potential interaction terms. One example would be in the field of medicine. If the accuracy of a diagnostic model to determine the onset of a heart attack could be improved by 0.01% by including an interaction term that was previously excluded, that represents a chance to save an additional life out of every 10,000 patients. With computing power growing cheaper and cheaper, such a tradeoff becomes increasingly worthwhile.

Another benefit to developing methods that automatically select interaction terms in Data Science problems is in providing further information to researchers about the problem structure. The larger and more complex the problem in terms of the number of potential variables, the less likely it is for researchers to have a complete understanding of the underlying dynamics. As such, the identification of even relatively minor interaction effects can provide

valuable direction for further research if there is no known theoretical basis for such effects. In the past, the main goal of variable selection methods was to find the minimum number of variables which provided the most information to the model, i.e. the focus was largely on the variables that contributed the most and weeding out variables that contributed relatively little. However, with model size no longer as limiting a constraint, perhaps we should expand our focus to exploring the variables which provide incremental benefits.

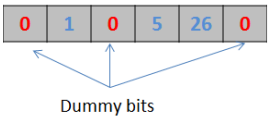
1.4 Including interaction terms in the GA framework

The biggest challenge in the inclusion of interaction terms in variable selection problems is the dramatic increase in the solution space. For now, we constrain ourselves to only considering second order interaction terms, i.e. only pair-wise interactions. For n main effects terms, this adds $\binom{n}{2}$ second order interaction terms. For relatively small n the additional terms can still be handled using the traditional GA variable selection chromosome (a single vector of 0-1 bits of length n+$\binom{n}{2}$ to indicate all possible variables), but this implementation quickly becomes unwieldy. For 100 variables an additional 4,950 interaction terms are added, and for 200 variables this jumps to 19,900. Thus for problems with hundreds of variables a new chromosome formulation is needed.

In order to solve the scalability issue, we propose some modifications to the original chromosome formulation. While only second order interaction terms are examined here, the basic technique for extending the GA framework remains applicable for higher order interactions at the cost of greatly increased computation time. Firstly, a maximum chromosome length l is defined. This allows the modeler to specify an upper bound for model sparsity, as in many instances modelers may not be interested in creating a model with thousands of variables. Secondly, instead of each bit in the chromosome simply being 0-1 to indicate the absence or presence of a variable, each bit now stores the index number of a variable to be included, and 0 if the bit is a “dummy bit”. Dummy bits are placeholder bits within the chromosome that reserve space for a potential variable to enter the model. This formulation allows for chromosomes representing models with a differing number of included variables

while still allowing chromosome length to be homogenous within the population, which simplifies the crossover operation.

Figure 1.4 : Chromosome with dummy bits



The chromosome in Figure 1.4 shows a chromosome of length 6 with 3 dummy bits, with variables 1,5 and 26 included in the model. Each new chromosome is initialized with dummy bits in all positions, and the number of initial variables is chosen uniformly between 1 and L (maximum number of variables). Pre-seeded variables can also be utilized instead of random selection. The index positions of these variables are also chosen by sampling without replacement from the available L positions, after which the variables (either randomly chosen or pre-seeded) are then filled into their respective index positions on the chromosome.

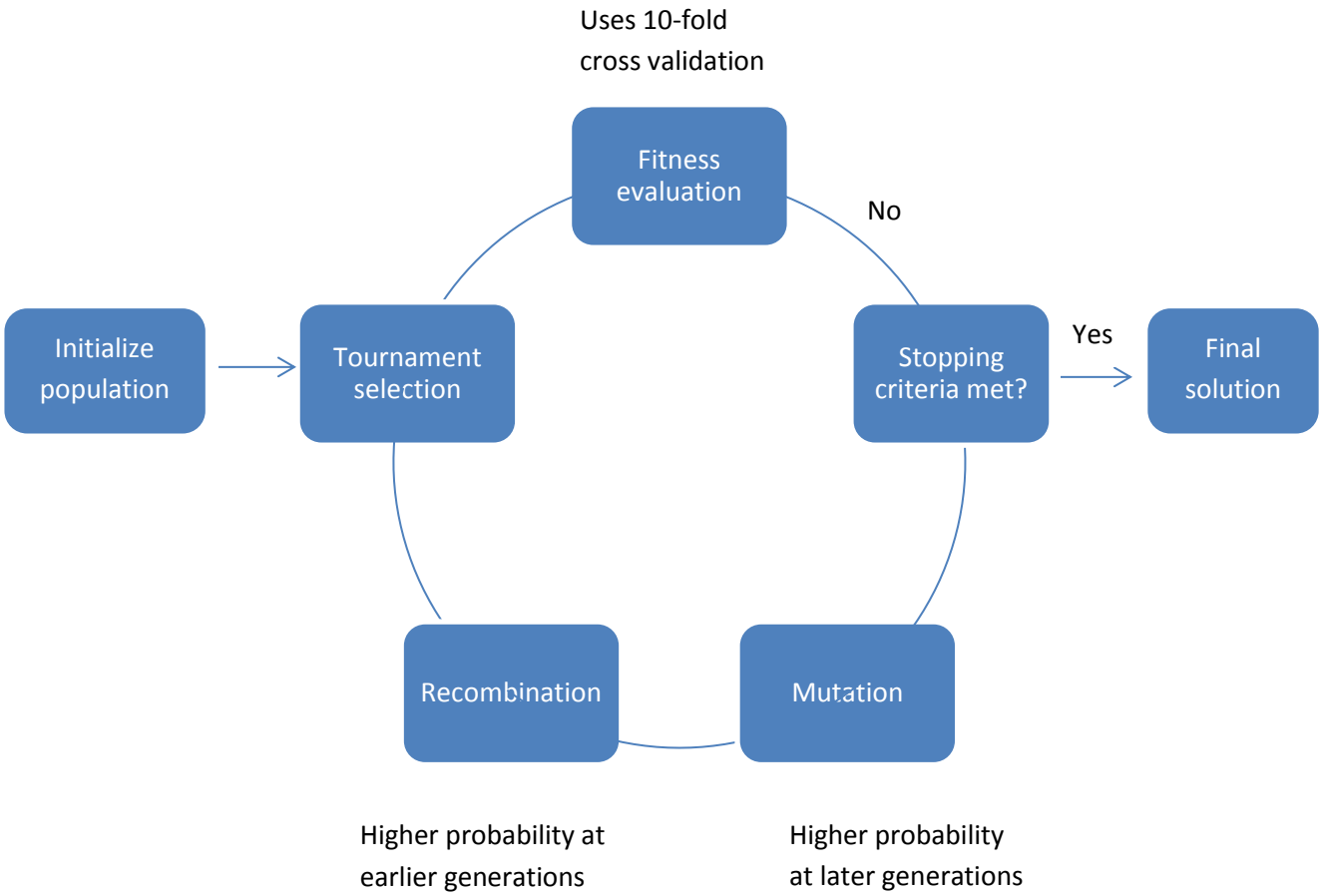
The current chromosome formulation can handle an arbitrary number of main effects terms in addition to interaction terms as long as the modeler specifies a maximum number of variables. As the chromosome length is homogenous, the aforementioned single point crossover operator still works on the modified chromosome, with some additional checks to ensure that duplicate variables are removed. However, the mutation operator now has to be separated into two types, a deletion mutation and an addition mutation. The deletion mutation replaces a random non-dummy bit with a value of 0, converting it to a dummy bit and removing the selected variable from the model. The addition mutation replaces a random dummy bit with a randomly selected variable that is currently not included in the model. Both types of mutation occur independently with probabilities P_a and P_d specified by the modeler. Both mutations occur simultaneously with probability $P_a * P_d$, resulting in one variable being switched out for another.

In addition, the GA framework ensures the model obeys strong hierarchy. Each time an interaction term enters the model through either recombination or the addition mutation, a

check has to be performed to ensure that the corresponding main effects terms are also included. If not, the missing main effects terms are inserted into random dummy bit positions. If a main effect term is deleted through the deletion mutation, then all interaction terms that include the aforementioned main effect term are also deleted.

Lastly, in order to prevent selection of models that over-fit the data, all fitness functions are evaluated using 10-fold cross-validation. The data is partitioned into ten folds, with the models being successively tested on a single fold and trained on the other nine folds. The final fitness is then obtained by averaging the model fitness over all ten test folds. With this process, there is never any overlap between data used for training models, and data used for evaluating the fitness. Figure 1.5 below outlines the high level process flow for the genetic algorithm variable selection framework.

Figure 1.5 : Genetic algorithm for variable selection process flow



1.5 Advantages and disadvantages of the GA framework

Many methods already exist to perform variable selection, some of which are able to handle the inclusion of interaction terms for datasets of varying sizes. Some examples of common methods are stepwise selection [1], random forests [4] and lasso [5]. Each of these methods comes with their own pros and cons. For example, stepwise selection has been criticized for violating assumptions of certain statistical tests involved in the process [9], and that the process reports narrower confidence intervals [10] and lower p-values than it should [11].

The GA framework for variable selection has several advantages (and of course disadvantages) compared to some of the other methods. The first advantage is that the GA framework is independent of the fitness function, meaning that the same variable selection process can be used for a large variety of models and for a large variety of predictor variables, as long as the appropriate fitness function is provided.

The second advantage is that the fitness function can be easily tailored towards a particular metric of interest. Many other variable selection procedures use statistical measures to try to quantify the importance of a variable when deciding whether said variable should be included in the model. However, despite these statistical measures being based on valid theoretical foundations, they may not perfectly coincide with the performance measure being applied to the resulting model. For example, a modeler may choose to use AIC when performing variable selection for a model, but choose to evaluate the candidate models according to classification error or the area under the ROC (AUC). Thus, while the variables selected are perfectly valid, the candidate models may not include the true best model (in terms of the chosen evaluation metric) as the selection procedure is not necessarily optimizing over the final model performance measure. For example, consider a model being developed for medical diagnosis of a life-threatening disease to which a statistical variable selection procedure is applied. The modeler could be interested primarily in the true positive rate of the classification model, and thus wish to find a model that maximizes the true positive rate (TPR). As the TPR is not a parameter that is being directly optimized by any statistical procedure, it is possible that there exists a model outside of the set of candidate solutions returned by the statistical variable

selection procedure that has a higher TPR, which therefore would be preferred by the modeler over any of the currently obtained candidate solutions. Using a GA to perform variable selection would allow the modeler to choose a combination of variables that result in higher TPR by specifying the model TPR as the fitness function, allowing for a direct link between the variable selection process and the final performance metric.

Lastly, the GA framework is inherently able to handle complex problem structures as it does not rely on any assumptions regarding the problem space, unlike some other methods. Methods which involve quantifying the marginal benefit of each variable work best when the variables being considered are not correlated and can be considered independently. However, when the problem space is complex and include significant interactions, it can be difficult for such algorithms to find global optimums which may only be found multiple "steps" away from local improvements.

That said, using a GA also comes with some disadvantages. First and foremost is that the GA is a search heuristic that does not have any theoretical guarantees of convergence or optimality. GA's have been shown to be able to find optimal solutions in many test applications. However, for problems in which the optimal solution is unknown there is no real way to validate if the GA solution is indeed optimal. There is also no way to determine if the obtained solution is globally or locally optimal. However, as the candidate solutions present in the population are all valid at each generation, the resulting best solution is also valid despite non-guaranteed optimality. Thus, as long as the resulting solution has a satisfactory fitness level it can be said that the GA has found a good solution. If optimality is not a primary concern (and for many Data Science problems it can be difficult to find optimal solutions due to the scale of the problem) then a GA can be utilized successfully despite not being guaranteed to converge to a global optimum solution.

The second disadvantage of the GA approach is that it tends to have a long run time compared to other methods. In some of the test sets which our GA was implemented on, the run time was measured in days. Part of the issue is that there is a lot of room for improvement in terms of optimizing the GA code we used. However, it is undeniable that a GA is a very computationally

intensive approach that may not be feasible when response time is an important consideration. This disadvantage has lately been mitigated by the increasing availability of cheap computing power. The GA approach also naturally lends itself easily to parallelization. Each member of the population can be evaluated individually, and all recombination and mutation operations can be easily performed in parallel as they do not affect any other members of the population. Thus, despite the run time being much longer for smaller scale problems, the GA is able to handle (albeit fairly slowly) very large scale problems that would cause memory issues for some other methods. The GA can also be terminated early if no improvement has been found for a certain number of generations, thus potentially saving some computation time. However, there is always the risk that the GA could have been caught in a local optima and was terminated before it could break out and find a better solution.

1.6 Experimental runs

The GA above was applied to several data sets to evaluate the benefits of including interaction terms. The statistical package R was used for the main GA code, with the RWeka package [12] being used to evaluate several fitness functions. The RWeka package allows the use of many types of machine learning models found in Weka [13], an open source machine learning software package, which increases the flexibility of the GA framework. Experimental work done so far has focused on logistic regression models, using either classification accuracy or area under the ROC curve as the fitness function. Tournament selection was used for all experimental runs (with the highest fitness solution being preserved), along with single point crossover. However, alternative selection methods should be explored for each application of the GA framework to a new dataset as there is no theoretical guarantee that tournament selection is the optimal selection method.

Both crossover and mutation (addition and deletion) are applied with a variable probability across the GA's run time. A minimum and maximum probability is defined for each operator (the same parameters apply to both types of mutation). The probabilities for each operator are adjusted each generation so as to vary from minimum to maximum or vice versa. The crossover probability p_c is initialized to $p_{c\_max} = 0.5$ at generation 0, and then varies across each

generation i according to Equation 1.3 below until finally reaching p_{c_min} = 0.2 after *maxgen* iterations.

$$p_c(i) = p_{c\_max} - \left(\frac{i}{maxgen}\right)\left(p_{c\_max} - p_{c\_min}\right) \quad (1.3)$$

The mutation probability p_m (for both addition and deletion) is initialized to p_{m_min} = 0.01 and varies linearly throughout the run until it reaches p_{m_max} = 0.2 after *maxgen* iterations, as shown in Equation 1.4.

$$p_m(i) = p_{m\_min} + \left(\frac{i}{maxgen}\right)\left(p_{m\_max} - p_{m\_min}\right) \quad (1.4)$$

These varying probabilities are chosen to obtain a higher chance of crossover with less mutation at the beginning of the GA run (increased exploration of solution space), and a lower chance of crossover with more mutations at the end of the run (increased exploitation of good solutions in population).

1.6.1 Simulated data

To first verify the ability of the modified GA framework as a search methodology, a simulated test dataset was created. The test dataset consisted of 100 rows of 3 variables (X_1, X_2, X_3), with $X_i \sim N(0,1)$ for i=1,2,3. A variable Y was created such that $y = x_1 + x_2 + x_1 * x_2$, with the predictand being a factor (class) satisfying the following condition:

$$class(y) = \begin{cases} 0, & y < 2 \\ 1, & y \geq 2 \end{cases}$$

A logistic regression model was created as a fitness function for the GA. The new GA framework was able to select the correct variables in less than 20 generations, demonstrating that the methodology is indeed able to determine the globally optimum solution.

1.6.2 Flu dataset

A second small dataset was used to evaluate the efficacy of the GA framework. The flu shots data [14] consists of 159 patients with 3 predictor variables (age, health awareness, and sex). Age and health awareness are numeric discrete variables, while sex is a binary factor (0

representing females and 1 representing males). The predictand is a binary factor indicating whether the patient received a flu shot (1 representing yes and 0 representing no). The data is first pre-processed by centering the age and health awareness variables.

Utilizing stepwise selection (using AIC) on the data yields a model with only age and awareness as significant predictors. The GA also returns the same model if AIC is used as the fitness function. However, if the AUC is used as the fitness function the GA returns a different model. The AUC of the model returned by stepwise selection is 0.763. By optimizing over AUC, the GA returned a model with awareness, sex, and awareness:sex as predictors with an AUC of 0.803. While sex and awareness:sex were not significant predictors based on p-values at a 0.05 significance level, the performance of the model based on cross-validated AUC was better (conversely, AIC was slightly higher for this alternative model). This can be seen as an example where the GA approach is highly flexible due to the customizable fitness function and is able to open up more choices for the modeler to consider. If the improvement in performance (measured by AUC) is not judged to be significantly higher, the modeler may prefer a model that may be more generalizable in that the predictors have better statistical properties. However, if there is a marked improvement in performance then the modeler may choose to utilize the GA solution despite the higher p-values of the predictors, especially if concerns of overfitting data have already been addressed through cross-validation or other methods.

References

[1] Efroymson,M. A. (1960) "Multiple regression analysis," Mathematical Methods for Digital Computers, Ralston A. and Wilf,H. S., (eds.), Wiley, New York

[2] Akaike, H. (1974), "A new look at the statistical model identification", IEEE Transactions on Automatic Control 19 (6): 716–723

[3] Mallows, C. L. (1973). "Some Comments on C_P". Technometrics 15 (4): 661–675

[4] Breiman, Leo (2001). "Random Forests". Machine Learning 45 (1): 5–32

[5] Ana Herrero, M. Cruz Ortiz (1999), “Qualitative and quantitative aspects of the application of genetic algorithm-based variable selection in polarography and stripping voltammetry”, Analytica Chimica Acta, Volume 378, Issues 1–3, 245-259

[6] Olivier Gayou, Shiva K. Das, Su-Min Zhou, Lawrence B. Marks, David S. Parda, Moyed Miften (2008), “A genetic algorithm for variable selection in logistic regression analysis of radiotherapy treatment outcomes”, Medical Physics 35, 5426

[7] John H. Holland (1975), “Adaptation in Natural and Artificial Systems”

[8] Stephanie Forrest, Melanie Mitchell (1993), “Foundations of Genetic Algorithms 2”

[9] Thompson, B. (1995). Stepwise regression and stepwise discriminant analysis need not apply here: A guidelines editorial. Educational and Psychological Measurement, 55, 525-534.

[10] Altman, D. G. and P. K. Andersen. 1989. Bootstrap investigation of the stability of a Cox regression model. Statistics in Medicine 8: 771-783.

[11] Wilkinson, L. (1979). Tests of significance in stepwise regression. Psychological Bulletin, 86, 168-174.

[12] Rweka, http://cran.r-project.org/web/packages/RWeka/index.html

[13] Weka 3: Data Mining Software in Java, http://www.cs.waikato.ac.nz/ml/weka/

[14] http://www.unf.edu/~jgleaton/LogisticRegressionFluShots.doc

Chapter 2 - Searching for interaction terms in high-dimensional datasets : an application of a GA framework for variable selection

2.1 Introduction

Much work has been done in developing variable selection techniques to help develop informative models. However, the detection and exploitation of interaction terms has been relatively neglected in these studies. In most cases, interaction terms are only included if the modeler has prior knowledge that certain interactions are expected to be significant predictors.

Part of the reason may be that interaction terms are often only marginally beneficial compared to main effects terms. In the interests of model sparsity, such terms may not be deemed worth the tradeoff of including additional predictors into the model. Furthermore, many commonly-used variable selection methods either do not include interactions or do not scale well with the inclusion of interaction terms. This is because interaction terms cause the potential predictor space to expand in a combinatorial fashion. With the increasing prevalence of very large data sets, this increase can be upwards of tens of thousands of additional variables. Compounding the issue is a lack of a definitive methodology to detect interaction terms. Most modelers rely on data visualization and domain expertise to identify possible interaction terms. With tens of thousands of variables, such methods become harder to implement and there is a greater possibility of missing out on potentially beneficial terms.

However, in recent years the rapid growth in available computing power as well as the increasing prevalence of very large rich datasets has changed the traditional approach to predictive modelling. Previously, modelers had to work with limited datasets from which they tried to build simple models by extracting variables with the most predictive power. We now have datasets large enough to build larger, more complex models with less fear of over-fitting and computers powerful enough to handle larger models and more complex algorithms. This allows modelers to focus more on data exploration and incremental improvements in model performance.

Our previous work in Chapter 1 introduced a variable selection framework utilizing a genetic algorithm (GA) that is designed to be able to deal with the inclusion of interaction terms in high-dimensional datasets. The GA framework provides several advantages over some of the more commonly used variable selection methods. Firstly, the GA can be modified to use any quantitative measure as the fitness function, rather than being tied to a certain measure such as AIC, BIC, area under the ROC etc. This allows the modeler to directly optimize over any measure of interest, instead of performing variable selection according to one measure and evaluating model performance based on another. Secondly, the GA formulation is inherently scalable. The number of potential predictors increases the size of the solution space, but other than perhaps increasing the required number of iterations or population size the basic methodology does not differ. This is in contrast to greedy algorithms which try to compute the marginal benefit of each potential predictor in isolation, an approach which may not be appropriate for very high dimensional problems with complex solution spaces. Lastly, the GA formulation is not a strictly greedy algorithm, compared to methods such as stepwise selection or random forest's variable importance metric. As such, the GA is more likely to be able to find solutions that are not local optima even when the solution space is relatively complex.

However, it should be noted that the final solution returned by the GA should not be treated as a definitive "optimal" solution. Just as with many other variable selection methods, the efficacy of the GA can vary depending on the structure of the dataset it is applied to. However, the GA framework's flexibility allows it to be easily applied to a wide variety of datasets and models. While the GA's optimal parameters (population size, number of generations, mutation and crossover probabilities) are difficult to determine theoretically, the GA's performance in terms of model fitness is fairly robust to these parameter settings as well, with the major difference being the run time (or number of generations needed to achieve similar fitness levels). In other words, if computation time is not a major concern the GA can be allowed to run for an arbitrarily large number of generations and obtain good results without needing to optimally tune the GA's parameters for each dataset specifically. Thus, the GA solution can be used as a good starting point for complex problems which are difficult to solve optimally, after which the insights obtained from the GA solution can be used to further refine the model.

Our initial study analyzed the GA framework's performance using a simulated small dataset and a small toy problem as a proof of concept. The GA was able to find the same solution as other established variable selection methods, while retaining the flexibility to optimize over a different performance measure if desired. To test the efficacy of the GA in a practical setting, the GA variable selection framework was applied to logistic regression models built on two real-world medical datasets.

2.2 Heart arrhythmia dataset

In order to evaluate the efficacy of a genetic algorithm for variable selection in a large dataset with potential interaction terms, the methodology was applied to a heart arrhythmia dataset from the UCI Machine Learning Repository [1]. The dataset contains 279 attributes (mostly ECG readings) with 452 observations. The predictors consist of a mix of categorical, real and integer variables. Each observation also has a classification ranging from 1 to 16. Class 1 indicates normal ECG, classes 2-15 indicate various classes of arrhythmia and class 16 is unclassified.

For this experiment, classes 2-16 were combined to transform the problem into a binary classification problem (normal vs abnormal). Out of the 452 observations, there were 207 normal patients and 245 patients with some form of arrhythmia. Additional pre-processing was performed on the data to remove attributes where more than 90% of the observations had a value of 0, and missing values were replaced by the medians. From there, the predictors were centered and scaled in order to reduce multi-collinearity effects.

The resulting dataset contained 261 main effect predictors (all numeric) along with the binary classification attribute. The addition of all possible 2-factor combinations of the predictors resulted in 33930 additional predictors being added to the solution space. While many of these interaction terms may not be informative, the sheer number of potential predictors presents difficulties for using data visualization techniques and/or expert judgement to pre-select interaction terms to be included for consideration. However, in order to even begin utilizing some of the more common variable selection techniques, a significant amount of data pruning would first have to be done to try to reduce the dimensionality of the problem. In this instance,

the statistical package R ran out of memory attempting to perform forward selection on a logistic regression model with all 34,191 main effects and pair-wise interaction terms. Similarly, backwards selection could not be performed as the full model could not be created. While data pre-processing is a necessary step in almost every modelling process, the larger the potential number of predictors the larger the possibility for unknowingly leaving out predictors that may provide some benefit in model performance despite appearing to be insignificant.

The area under the ROC curve (AUC) was chosen as the primary metric for model performance in the GA as it is a measure that is widely used and understood in the medical community.

2.2.1 GA variable selection

The GA framework was applied using a logistic regression model to the same data for 5 runs with different initial random seeds. For each run, a population size of 30 was used, with each candidate solution having a minimum of 5 predictors and a maximum of 100. The GA was set to terminate after 250 generations to allow sufficient time to find good solutions. Although the GA has a termination clause to stop the algorithm after a predetermined number of generations without improvement in the best fitness value, in this case the termination clause was disabled as the long run time is not a concern and we wish to maximize the possibility of finding a globally optimal solution.

To achieve a good mix of recombination and mutation, the recombination and mutation probabilities were set to vary as each run progressed, with a high starting probability of recombination ($p_{combine} = 0.5$) and a low starting value of mutation ($p_{mutate} = 0.01$). These values allow for a higher chance of exploration at the initial stage of the GA. As the GA nears the maximum number of generations, the recombination probability is decreased gradually to $p_{combine} = 0.2$ and the mutation probability is increased gradually to $p_{mutate} = 0.2$. The higher chance of mutation and lower chance of recombination allows for more exploitation of existing good solutions, rather than introducing new untested solutions.

After 5 runs, the best solution found after 400 generations by the GA had an AUC of 0.922 with an associated 95% confidence interval of [0.896, 0.947] which is comparable to the stepwise

and random forest solutions. The GA solution contained 45 main effects predictors and 10 interaction terms, of which 2 had significant p-values in the logistic regression model.

Figure 2.1 : GA maximum fitness values over 400 generations

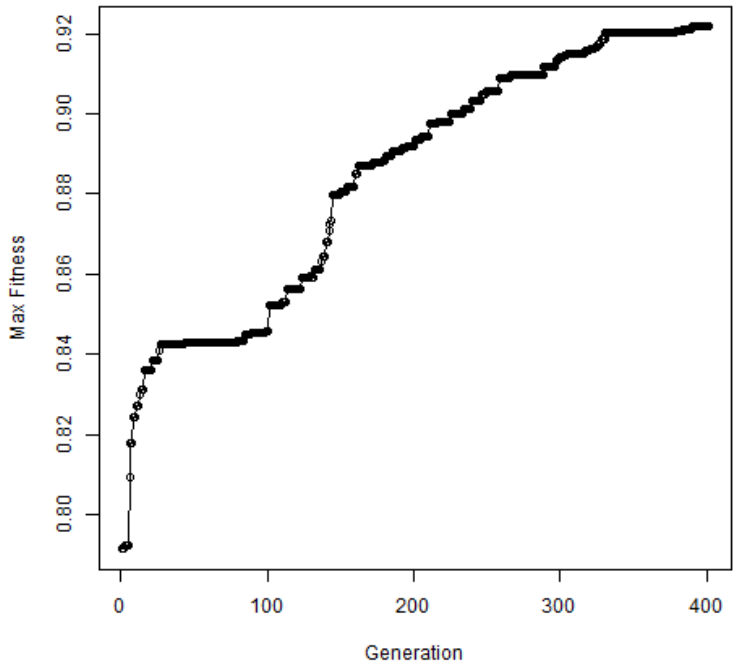


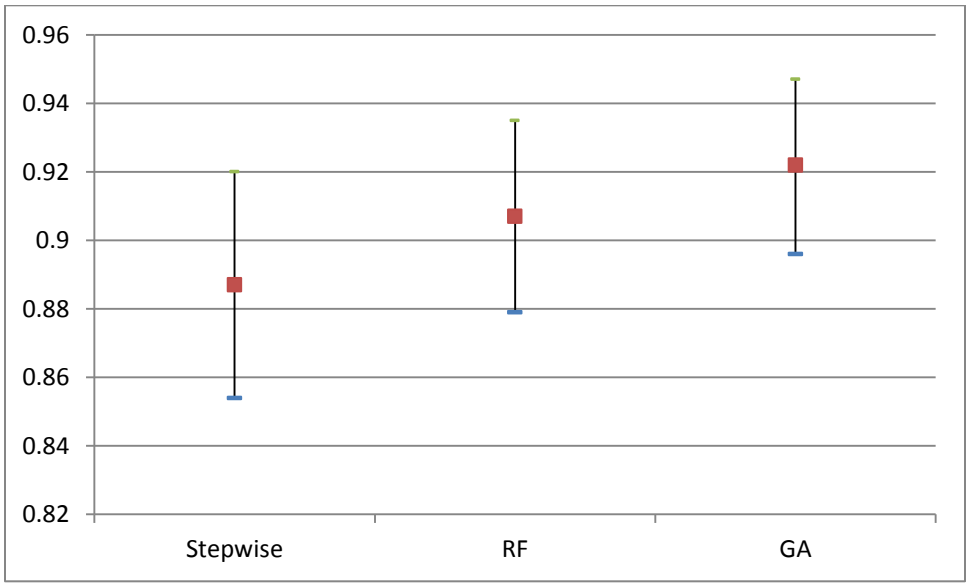
Figure 2.1 shows the change in the maximum fitness values across generations for the best GA solution. It can be seen that there are much larger jumps earlier on due to the increased probabilities of recombination resulting in a larger variety of solutions. Despite becoming relatively stagnant between approximately generation 30 to generation 90, the GA is eventually able to find markedly improved solutions and improve the AUC to 0.922. It must be noted that there is no guarantee that 0.922 is the global optimum and increasing the run time could result in further improvements in the AUC. However the purpose of this study is to investigate the feasibility of the GA framework on a large dataset with interaction terms, and not to find a specific solution with a globally maximal AUC. Thus, the solution obtained was deemed acceptable for our purposes.

For comparison purposes, two other models were built and evaluated based on AUC. The first is a logistic regression model using stepwise selection with only main effect variables included in

the model as R was unable to handle the inclusion of all possible interaction terms. The main effects model contained 39 predictors after stepwise selection, with 23 of those predictors having p-values below 0.05. The AUC obtained using 10-fold cross-validation was 0.887 with an associated 95% confidence interval of [0.854, 0.920]. For a second comparison, a random forest was built using the randomForest package [2] in R with 500 trees. The random forest model performed slightly better than stepwise selection with only main effects, obtaining an AUC of 0.907 with an associated 95% confidence interval of [0.879, 0.935].

Figure 2.2 summarizes the mean cross-validated AUC for the three methods outlined above, along with the bounds of the 95% confidence intervals. All three methods result in comparable AUCs as the confidence intervals overlap, although the GA solution generally performs better. As expected the model returned by stepwise selection using only main effects possesses the lowest mean AUC.

Figure 2.2 : Mean cross-validated AUC and 95% confidence intervals for stepwise selection, random forests and GA selection



While the AUC gives a useful measure of a model's performance over various thresholds, in many medical datasets the true positive rate is also of great import to modelers. When considering a potentially life-endangering condition, the consequences of a false negative (or

alternately speaking, missing a true positive case) are likely worse than a false positive. Figure 2.3 summarizes the confusion matrix for all three models. While the GA fitness function is not explicitly optimizing for a higher true positive rate (although this is also possible using the flexibility of the GA framework), the GA solution in this dataset provides a balance between a high AUC as well as a high TP rate.

Figure 2.3 : Confusion matrix for stepwise, random forest and GA selection

	TP	TN	FP	FN	ACC	TPR	TNR	PPV
Stepwise	226	173	34	19	0.8827	0.9224	0.8357	0.8692
RF	208	159	48	37	0.8119	0.8490	0.7681	0.8125
GA	222	176	31	23	0.8805	0.9061	0.8502	0.8775

The results show that the GA framework performs fairly well on the arrhythmia dataset with minimal tuning and data pruning needed. It was able to handle a large expanded solution space with the inclusion of interaction terms and find solutions with a high AUC and acceptable true positive and true negative rates and the best positive predictive value, while not being subject to the criticisms against stepwise selection procedures. The random forest solution also performed well in terms of AUC, but has a poorer true positive and true negative rate, although this could be improved by further refining the random forest parameters. However, while the random forest solution is also able to handle interaction terms, it is also less interpretable as the interaction effects are encoded in the tree structure instead of explicitly defined as in a logistic regression model. This leads to random forest models being harder to use for data exploration.

2.3 Physionet MIMIC II Clinical database

The GA framework was also applied to a second dataset obtained from PhysioNet [3]. The MIMIC II clinical database contains clinical records for 4000 patients in ICUs. The dataset contains a total of 44 mostly numeric predictors (3 factor predictors) with the predictand being death. However, not every predictor is included for each patient. Furthermore, the clinical records for each patient can contain repeated measures where a clinical measurement is

performed over time. Out of the 4000 patients, 554 patients eventually passed away while the remaining 3446 survived.

In order to encapsulate the information from the repeated measures into a single numeric variable, for each predictor the median, standard deviation and 25^{th} and 75^{th} percentiles were calculated and used as predictors. These predictors were chosen to provide information on the average value of each measure, the variability in the measurements, as well as the extremes of the measurements. As part of our study, we also examined the use of entropy as a summary statistic of complexity. We compared sample entropy and permutation entropy against the runs test and permutation test for randomness on a variety of datasets and found that in many cases the entropy measures and tests for randomness ranked the test datasets in the same order of complexity (high degree of randomness being considered equivalent to low complexity, and vice versa). In particular, there was a high degree of similarity between permutation entropy and the permutation test in terms of both theoretical formulation as well as experimental results. The paper is included in Appendix 2A. Measures such as sample entropy, permutation entropy or the permutation test are possible alternatives to handle repeated measures such as those found in the MIMIC II dataset.

Missing data was replaced with median values and the data was scaled by subtracting the mean. This resulted in 146 main effect predictors. The inclusion of all pairwise interaction terms results in an additional 10585 variables.

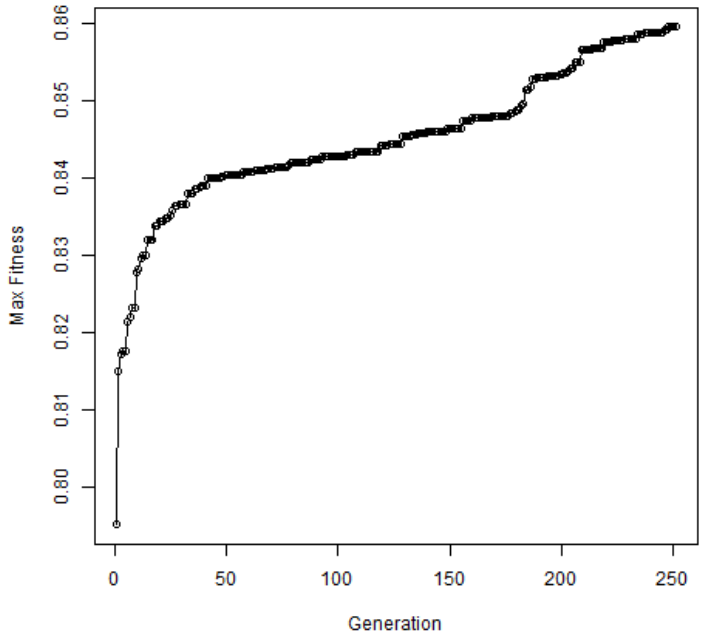
2.3.1 GA selection

The GA framework was applied to the MIMIC II dataset using a logistic regression model with AUC as the fitness function. Once again, 5 runs were performed using different seeds. The GA parameters used were similar to those used for the arrhythmia dataset, with a population size of 30 and each candidate solution having a minimum of 5 predictors and a maximum of 100. 250 generations was chosen as the termination criteria.

Likewise, the recombination and mutation probabilities were again varied throughout the length of each GA run, with $p_{combine}$ ranging from 0.5 to 0.2 and p_{mutate} ranging from 0.01 to 0.2.

Figure 2.4 shows the change in maximum fitness over the generations for the best found GA solution. The final AUC obtained was 0.859, with a 95% confidence interval given by [0.844, 0.874]. The logistic regression model contained 52 main effects terms and 30 interaction terms (of which 21 had p-values < 0.05).

Figure 2.4 : GA maximum fitness over 250 generations



Similar to the arrhythmia dataset, a stepwise selected logistic regression model and a random forest model was used as a comparison with the GA selected model. In this instance, R was able to perform stepwise selection on the full model including interaction terms. The resulting model contained 29 main effects terms and 74 interaction terms, with an AUC of 0.839 and a 95% confidence interval of [0.820, 0.859]. For the second model, a random forest was used with 500 trees. The resulting model had an AUC of 0.833 and a 95% confidence interval of [0.817, 0.850].

Figure 2.5 : Mean cross-validated AUC and 95% confidence intervals for stepwise selection, random forests and GA selection

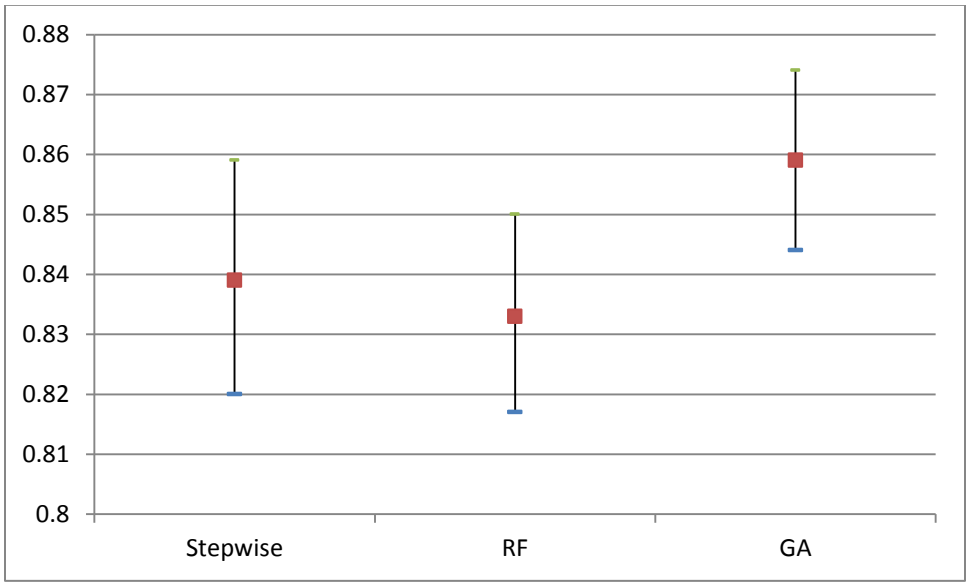


Figure 2.5 above summarizes the mean cross-validated AUC and associated 95% confidence intervals for the three models evaluated. The GA solution performed slightly better when evaluated using AUC. In contrast with the arrhythmia dataset, the logistic regression model from stepwise selection had a higher mean AUC than the RF solution. This is likely due to the smaller total number of variables in the MIMIC II dataset allowing stepwise selection to include interaction terms.

Figure 2.6 : Confusion matrix for stepwise selection, random forest and GA selection

	TP	TN	FP	FN	ACC	TPR	TNR	PPV
Stepwise	208	3366	80	346	0.8935	0.3755	0.9768	0.7222
RF	67	3409	37	487	0.8690	0.1209	0.9893	0.6442
GA	178	3345	101	376	0.8808	0.3213	0.9707	0.6380

A look at the confusion matrix for these models (shown in Figure 2.6 above) shows that the GA again provides a balance between mean AUC and a good true positive rate. The random forest model performed much worse in terms of the true positive rate on this dataset, correctly predicting only 67 out of 554 total deaths. The GA solution was also more parsimonious than

the solution presented by stepwise selection (82 vs 103 predictors respectively) and more easily interpretable than the random forest model in terms of data exploration.

2.4 Discussion

The GA framework was applied to two complex medical datasets with a large number of predictors. The GA selection procedure was able to handle the large number of interaction terms and generate solutions that were improvements (measured by AUC) over those obtained by stepwise selection and random forest. These results also demonstrate the advantage of the GA framework using a customizable fitness function, as the variable selection method is able to directly optimize the candidate model according to the same metric used for model evaluation, instead of using a proxy statistical measure of variable importance such as AIC etc. Most importantly, the GA was also able to identify interaction terms that provided some significant benefit to the model, which could be difficult to determine using standard variable selection methods if the set of potential predictors is large enough.

However, it should be noted that the GA selection procedure would be most effective when used in conjunction with other variable selection methods. As the GA does not penalize model complexity, the GA solution can tend to include extraneous variables. However, the GA solution can be used as a pruned subset of variables, upon which other variable selection methods (e.g. lasso for regression models) can be applied to further reduce model complexity. Alternative variable selection methods can also be used to generate good initial solutions for the GA, which could help guide the GA by starting out in high fitness areas in the solution space.

Thus, the main benefits of the GA approach to variable selection can be summarized as follows:

1) Scalability : The GA is able to handle datasets with an arbitrarily large number of predictors (including interaction terms) with minimal changes to its parameters, as long as run time is not a limiting factor
2) Flexibility : The GA is able to use custom fitness functions to better align the selection process with model evaluation, and the same framework can be applied to a variety of different models with different predictor types

3) Ease of use : The parameters used in the GA do not require extensive tuning, and the GA framework can be integrated with other variable selection methods to obtain better solutions

References

[1] UCI Machine Learning Repository, http://archive.ics.uci.edu/ml/datasets/Arrhythmia

[2] R randomForest package, https://cran.r-project.org/web/packages/randomForest

[3] PhysioNet , http://physionet.org/mimic2/

Appendix 2A

Comparing entropy with tests for randomness as a measure of complexity in time series

Chee Chun Gan cg8pa@viriginia.edu
Department of Systems and Industrial Engineering
University of Virginia

Gerard Learmonth jl5c@virginia.edu
Center for Leadership Simulation and Gaming
Center for Large-Scale Computational Modelling
Frank Batten School of Leadership and Public Policy
University of Virginia

Abstract

Entropy measures have become increasingly popular as an evaluation metric for complexity in the analysis of time series data, especially in physiology and medicine. Entropy measures the rate of information gain, or degree of regularity in a time series e.g. heartbeat. Ideally, entropy should be able to quantify the complexity of any underlying structure in the series, as well as determine if the variation arises from a random process. Unfortunately current entropy measures mostly are unable to perform the latter differentiation. Thus, a high entropy score indicates a random or chaotic series, whereas a low score indicates a high degree of regularity.

This leads to the observation that current entropy measures are equivalent to evaluating "how random" a series is, or conversely the degree of regularity in a time series. This raises the possibility that existing tests for randomness, such as the runs test or permutation test, may have similar utility in diagnosing certain conditions.

This paper compares various tests for randomness with existing entropy-based measurements such as sample entropy, permutation entropy and multi-scale entropy. Our experimental results indicate that the test statistics of the runs test and permutation test are often highly correlated with entropy scores and may be able to provide further information regarding the complexity of time series.

Keywords : entropy, sample entropy, permutation entropy, multi-scale entropy, tests of randomness, runs test, permutation test, time series complexity

1. Introduction

Entropy measures have gained widespread use in the analysis of complex real-world data. The term "entropy" first originated in the field of thermodynamics and can be interpreted as the amount of information needed to completely specify the physical state of a system. A very orderly and regular system has a low entropy value. An example of this is a system consisting of a container of hydrogen and helium molecules where all the hydrogen molecules are on one side of a divider and all the helium molecules are on the other side. In contrast, a system where the hydrogen and helium molecules are uniformly distributed throughout the container has very high entropy as the position of each molecule has to be completely specified in order to describe the state of the system.

The concept of entropy was further developed in the field of non-linear dynamic analysis and chaos as a measure of the complexity of a system. In Shannon's [1] seminal work on information theory, he defined entropy as the "information content" of a system. However, the concept of entropy remained largely theoretical until Pincus [2] developed Approximate Entropy (ApEn) as a measure of changing complexity which could be applied to real-world data sets. Following on from Pincus's work, various other entropy measures have been proposed for the same purpose. Richman and Moorman [3] introduced Sample Entropy (SampEn), a modified version of ApEn, to correct for the self-match bias in ApEn and to improve on several other statistical properties. Bandt and Pompe [4] proposed Permutation Entropy (PermEn) as an alternative measure of complexity for time series. Costa et. al. [5] developed Multi-Scale Entropy (MSE) to account for structural interactions across multiple time scales.

However, the abovementioned entropy measures all share a common attribute in that a maximal entropy score is assigned to completely random data, i.e. white noise. In that sense, entropy can be considered to be a measure of the degree of regularity in data where the presence of underlying structure will reduce the entropy score from the maximal value.

By extending on this premise, it should be possible to obtain similar information by utilizing existing statistical tests for randomness such as the Runs test and permutation test [6]. The rest of this paper explores the efficacy of such tests as compared to the abovementioned entropy measures. Section 2 outlines the basic framework of each entropy measure and test for randomness that was utilized in the experimental runs. Section 3 contains the experimental results for various test data sets, while section 4 discusses the conclustions from the experiments.

2. Experimental methods

2.1 Entropy measures

Sample Entropy(m,r)

As presented by Lake, Richman, Griffin and Moorman in their study of neonatal heart rates [7], SampEn is the negative natural logarithm of the conditional probability that a dataset of length m, given that it has repeated itself for m points (within a tolerance limit r that is commonly based on the standard deviation of the data), will repeat itself for m+1 points. SampEn can be calculated using the following equation:

$$SampEn = -log\frac{A}{B}$$

where A is the number of pairs of vector subsets of length m+1 which have a distance function less than r, while B is the number of pairs of vector subsets of length m which similarly have a distance function less than r. The main difference between SampEn and ApEn is that SampEn does not allow self-matching of points while ApEn does, meaning that ApEn always has a value of at least 1 for A and B.

For our experiments, the parameter m = 2 was chosen and r was set to be 0.2 times the standard deviation of the test data. A SampEn score of 0 indicates linear or highly regular data, while randomly generated data returns a SampEn score between 2.2 and 2.3.

Permutation Entropy(n)

PermEn was developed to handle the presence of noise in real-world data. For a time series $\{x_0, \ldots, x_{N-1}\}$ the PermEn algorithm splits the data into overlapping n-tuples, where n is the embedding dimension. Each n-tuple is then sorted in ascending order, which generates a "permutation type" π according to the ordering of the sorted data. As an example, consider the 3-tuple $\{x_0, x_1, x_2\} = \{3,5,1\}$. The sorted tuple is $\{x_2, x_1, x_0)$ which leads to a π = 2,1,0. For embedding dimension n there are n! possible permutation types. The relative frequency $p(\pi_i)$ is determined for each π_i , for $1 \leq i \leq n!$, according to the following equation:

$$p(\pi_i) = \frac{number\ of\ occurrences\ of\ type\ \pi_i}{N - n + 1}$$

The permutation entropy H(n) is then calculated as follows:

$$H(n) = -\sum_{i}^{n!} p(\pi_i)\log p(\pi_i)$$

H(n) ranges from 0 to log n!, with 0 indicating a series that is monotonically increasing or decreasing and log n! indicating a completely random series. In the experimental portion, H(n) is rescaled by dividing by log n!, thus normalizing H(n) to return values between 0 and 1 with 0 indicating highly regular data and 1 indicating maximal entropy. The parameter n = 5 was used for the calculation of H(n).

Multi-Scale Entropy (m,n)

In their paper [5], Costa, Goldberger and Peng observed that most entropy measures only consider a one-step difference and thus only measure entropy based on the smallest scale. Multi-Scale Entropy (MSE) down-samples the original time series $\{x_1, \ldots, x_N\}$ according to a scale factor T. The new time series $\{y^T\}$ is obtained using the formula $y_j^T = \frac{1}{T}\sum_{i=(j-1)T+1}^{jT} x_i\,, 1 \le j \le \frac{N}{T}$. In effect, the original series is partitioned into N/T disjoint sets. The mean of each disjoint set then forms a data point in the new series $\{y^T\}$. Sample entropy is then calculated while varying T and the resultant values plotted against T.

Since the MSE methodology revolves around the resampling process, in our experiments the resampling process was used before applying the various methodologies examined. Scale factors {2,3,4,5,10} were evaluated, with scale factor 1 being excluded as the down-sampled series would be identical to the original series.

2.2 Tests for randomness

Permutation test (t)

The permutation test for randomness [6] should not be confused with the permutation tests involving reshuffling of data to obtain more accurate test statistics. The permutation test for randomness is performed by first partitioning the original time series into groups of t elements. In the event that the original time series is not perfectly divisible by t, the remaining data points are discarded. The elements of each group are then sorted to obtain an ordering of the element indices. As there are $t!$ possible orderings in each group, a chi-square test can then be performed with t! categories and the assumption that the probability of each distinct ordering is $1/t!$. The chi-square statistic is interpreted as the distance from the expected value given the null hypothesis that the input data is uniformly distributed.

Thus, a high value indicates a high degree of regularity (conversely, departure from randomness) while a low value indicates a high likelihood of the null hypothesis being true.

It should be noted that the algorithm for the permutation test for randomness is very similar to that for calculating permutation entropy as presented in section 2.2. The main differences are that the partitions in the permutation test do not overlap, and that instead of summing the chi-square test statistic for each permutation the permutation entropy algorithm calculates H(n). In our experiments, a partition size of t=5 was chosen, in part to be consistent with the tuple size for permutation entropy.

Runs test

The runs test [6] examines the time series for the length of sequences that increase or decrease monotonically. The underlying basis for the runs test is that a non-random series will tend to have either more or less frequent runs than expected under a purely random distribution. For the purposes of our experimental analysis, the R function runs.test() in the "lawstats" package [8] was used to calculate the two-sided runs test statistic and p-values.

Similar to the permutation test statistic, the runs test statistic can also be interpreted as the distance from the expected value given a null hypothesis of a random originating distribution. A large difference from the expected value indicates a large degree of regularity in the series. Conversely, a difference close to zero indicates a high probability of the null hypothesis being true.

3. Experimental results

Entropy measures are primarily used to provide quantitative comparisons between various time series. By themselves, the various entropy scores are not sufficient to determine if a series is chaotic. However, entropy scores can be used to rank multiple time series according to the degree of regularity exhibited.

Conversely, tests for randomness have traditionally been used to provide probabilistic likelihoods of whether the input series is randomly distributed. Thus, the main focus of these tests has been on the p-values, or probability of encountering such an input series given the assumption that the null hypothesis is true (the series is randomly distributed) rather than the actual values of the test statistic. However, when used as a comparative measure among multiple time series the test statistic may also be able to give pertinent information, in a fashion similar to comparing entropy scores.

In our experiments, we evaluated the performance of Sample Entropy (SampEn), Permutation Entropy (PermEn), the permutation test chi-square test statistic (p.test) and the runs test statistic (runs.test). In certain cases, the Multi-Scale Entropy (MSE) framework was applied by downsampling the original series and evaluating the changes in the various metrics as the scale factor T increases.

3.1 Random data

The first test performed compared three time series (each with 1000 data points) generated from random distributions. The first series was generated from a Uniform(0,1) distribution, the second series from a Normal(0,1) distribution, and the third series from an Exponential(1) distribution. In each case, 30 replications of 1000 data points were generated and the mean score recorded. Figures 1a to 1c below show the resulting scores from SampEn, PermEn, the permutation test chi-square statistic and the runs test statistic. All scores for the various metrics are rescaled to be between 0 and 1 for better comparison. In addition, the inverse of the natural log is applied to the chi-square statistic from the permutation test, while the inverse of the absolute runs test statistic is shown. This was done to invert the plots to better correspond with the interpretation of the entropy measures, i.e. a high value corresponds to low regularity and a low value corresponds to high regularity. Detailed results of the tests, including p-values, can be found in Table 1.

Figure 1a

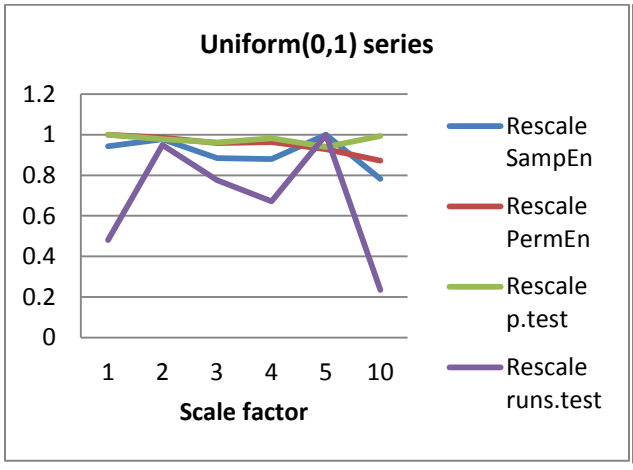


Figure 1b

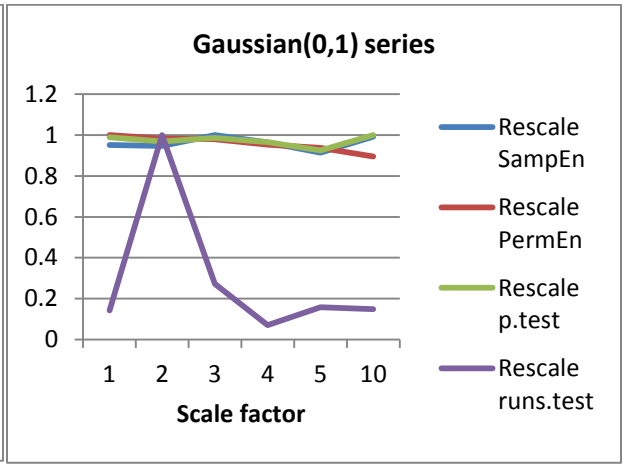


Figure 1c

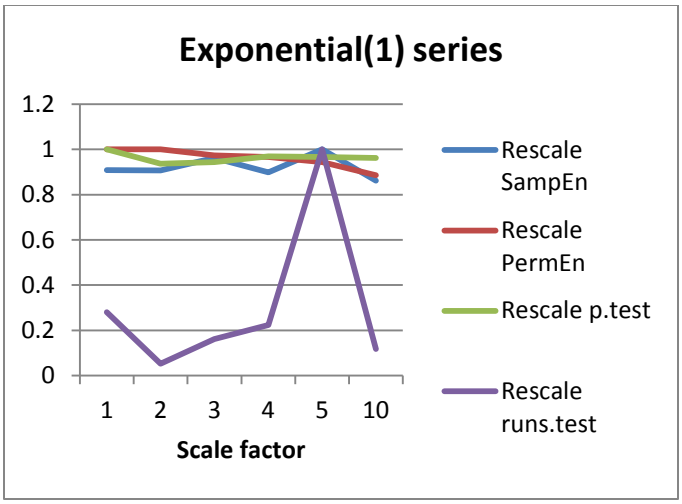


Table 1 : MSE analysis for random data

Uniform						
Scale factor	**1**	**2**	**3**	**4**	**5**	**10**
SampEn(2)	2.238808	2.32396	2.100495	2.089392	2.374906	1.856298
PE(5)	0.987112	0.973145	0.946222	0.951224	0.917653	0.860831
Perm test(5)	108.3935	120.7855	130.3399	117.9717	145.9562	111.9328
Perm test(5) p-val	0.7471	0.437094	0.224877	0.509409	***0.04717***	0.664233
Runs test	-0.8859	-0.44766	0.547999	-0.63373	-0.42533	-1.80916
Runs test p-val	0.3757	0.654397	0.583693	0.526258	0.670593	0.070426
Normal						
Scale factor	**1**	**2**	**3**	**4**	**5**	**10**
SampEn(2)	2.168564	2.155924	2.27835	2.197225	2.083466	2.261763
PE(5)	0.988476	0.971931	0.968727	0.942845	0.92731	0.884947
Perm test(5)	117.9929	130.3844	119.4328	132.3682	163.9508	111.9328
Perm test(5) p-val	0.5089	0.224066	0.471603	0.189728	***0.003987***	0.664233
Runs test	-0.6328	0.089532	-0.3288	-1.26746	-0.56711	-0.60305
Runs test p-val	0.5269	0.928659	0.742307	0.204991	0.570638	0.546473
Exp						
Scale factor	**1**	**2**	**3**	**4**	**5**	**10**
SampEn(2)	1.669779	1.665106	1.768694	1.650423	1.836711	1.581786
PE(5)	0.98541	0.985588	0.959452	0.953136	0.930561	0.873627

Perm test(5)	93.9944	127.9846	123.0685	108.374	109.967	111.9328
Perm test(5) p-val	0.9561	0.270497	0.380607	0.74757	0.711414	0.664233
Runs test	-0.5062	-2.68597	-0.8768	0.633729	-0.14178	-1.20611
Runs test p-val	0.6127	***0.007232***	0.380596	0.526258	0.887255	0.227776

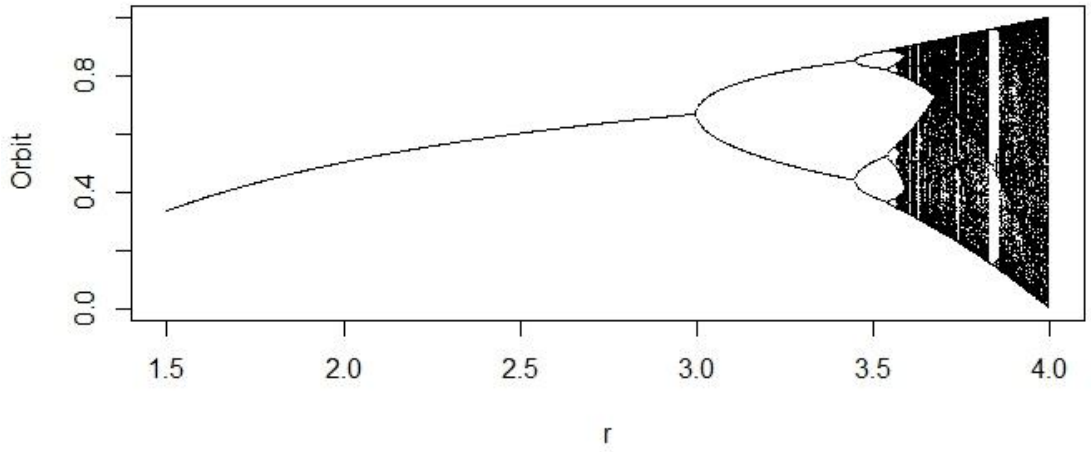
It can be seen that both PermEn and the permutation test show very high degrees of randomness for all three series. For the exponential series, while the rescaled SampEn score is consistently high through all scale factors, the absolute score shows a decrease in complexity when compared to the Uniform and Gaussian series. In most cases the p-values for both the permutation test and the runs test are well above common significance thresholds, correctly indicating that these series cannot be rejected as being random. However, there are several outliers highlighted in Table 1 where the p-values are below 0.05, which would lead to a rejection of the null hypothesis at a 0.05 significance level. In particular, there is significant variation in the runs test statistic compared to the other three test metrics.

We were unable to replicate the results of Costa et. al. [5], which found that the SampEn score monotonically decreases as T increases for series containing pure white noise. For SampEn, the permutation test statistic and the runs test statistic there was no discernible trend for varying T. However, the PermEn score did decrease monotonically as T increased.

3.2 Logistic Map

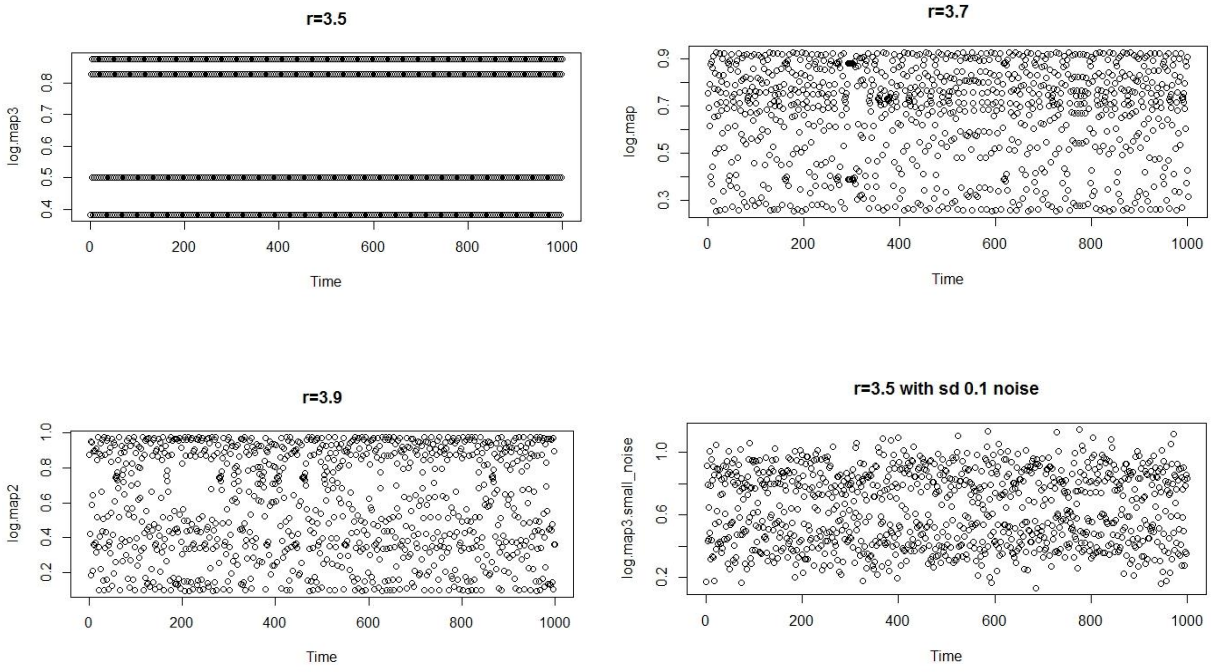
The second set of time series used was generated using the logistic map $x_{n+1} = rx_n(1 - x_n)$. The orbit of the logistic map is shown below in Figure 2 for values of r ranging from 1.5 to 4. Five separate time series were generated from the logistic map. The first three series consist of 1000 points generated using x=0.3, with r taking values 3.5, 3.7 and 3.9 respectively. To remove transient effects, 5000 points were generated and the last 1000 used for the analysis.

Figure 2 : Logistic map for increasing r



As can be seen from Figure 2, with r=3.5 the series has period 4, while for r=3.7 and r=3.9 the series results in deterministic chaos and appears much more random. Theoretically, the series with r=3.9 should exhibit a higher degree of chaos compared to the series with r=3.7. For the last time series in the set, the data from the periodic series with r=3.5 was used as the base and a random Gaussian(0,0.1) noise was added to each of the points. The plots of these four time series are shown in Figure 3.

Figure 3 : Time series from logistic map



The same scores from the preceding section were calculated for these 4 time series, with the same parameter values. Figure 4 shows the rescaled scores while the raw data is provided in Table 2.

Figure 4 : Comparison for logistic map

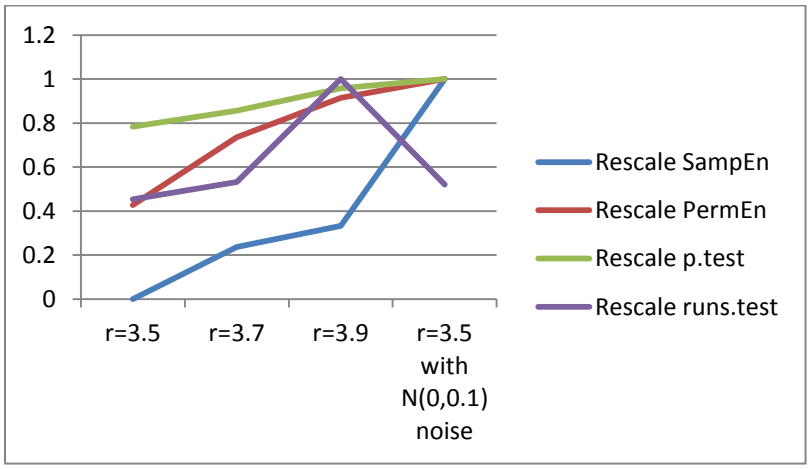


Table 2 : Scores for logistic map

	r=3.5	**r=3.7**	**r=3.9**	**r=3.5 with N(0,0.1) noise**
SampEn(m=2)	0.0000	0.3479	0.4883	1.4431
PermEn(5)	0.2896	0.4978	0.6185	0.6781
Perm. test(5)	5799.6520	2781.8331	1200.3280	936.3438
Perm. test(5) p-val	0.0000	0.0000	0.0000	0.0000
Runs test	31.5753	26.9561	14.3252	28.2849
Runs test p-val	0.0000	0.0000	0.0000	0.0000

SampEn, PermEn and the permutation test chi-square statistic are all able to detect the increase in chaos in the series. The runs test statistic does not display as much consistency as the other 3 measures, but the p-values for both the permutation test and the runs test enable us to easily reject the null hypothesis of a random process for all four series.

The MSE framework was then applied to the data sets with r=3.7 and r=3.5 with Gaussian(0,0.1) noise to evaluate the effects of increasing scale factor. Figure 5 shows that all four metrics generally vary in the same fashion for increasing scale factor.

Figure 5

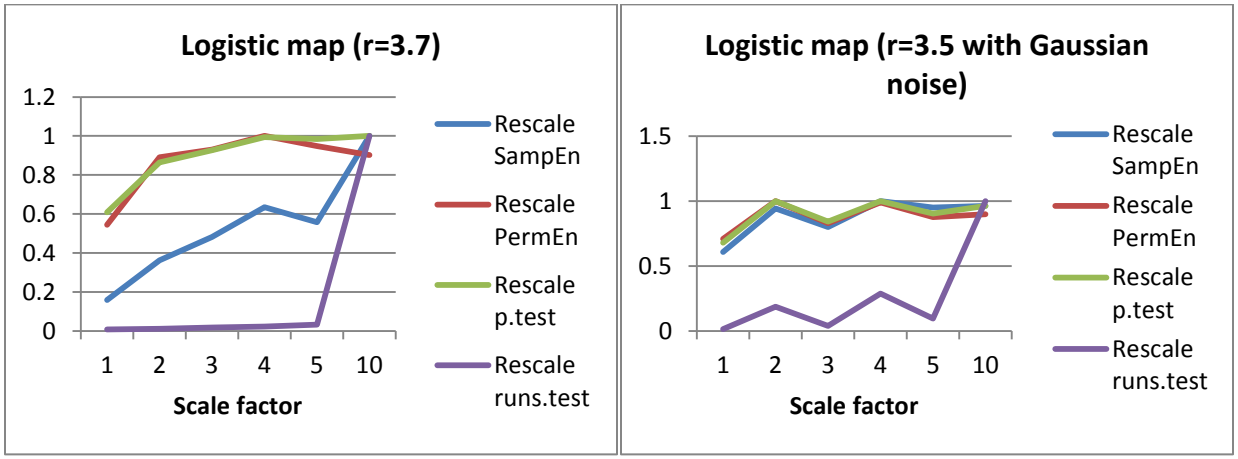


Table 3: MSE analysis for logistic map

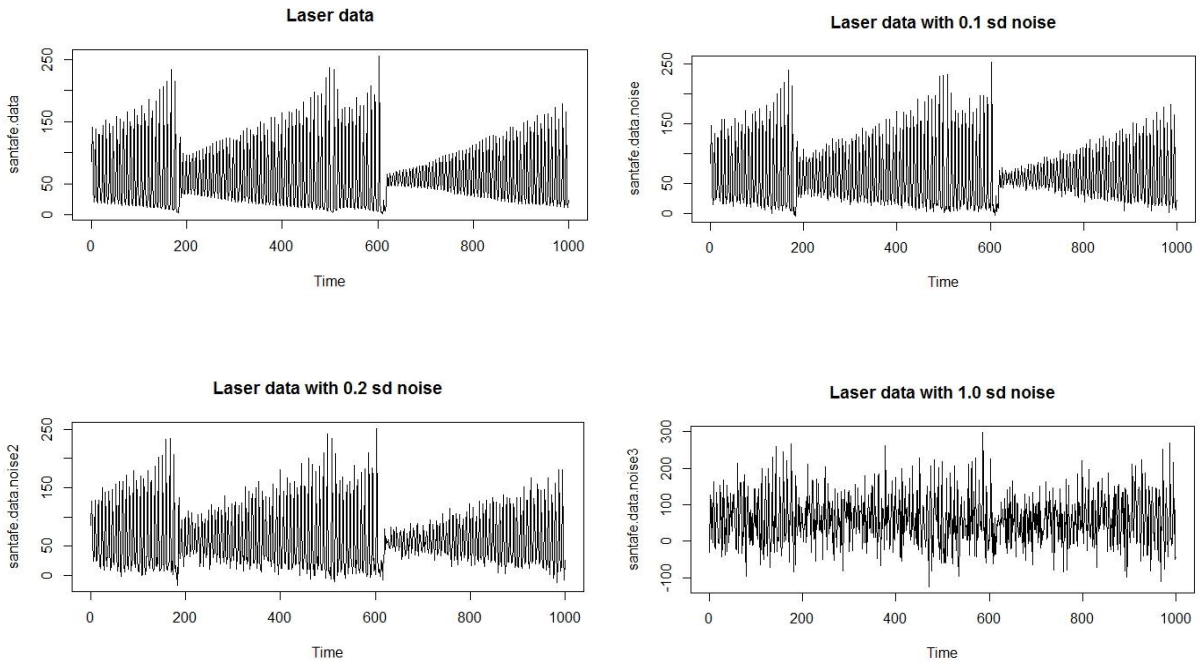
Logistic map r=3.7						
Scale factor	**1**	**2**	**3**	**4**	**5**	**10**
SampEn(2)	0.3479	0.7899	1.0515	1.3852	1.2181	2.1832
PE(5)	0.4978	0.8134	0.8494	0.9134	0.8667	0.8239
Perm test(5)	2781.8331	262.3685	181.2398	127.5694	133.9598	123.9256
Perm test(5) p-val	0.0000	0.0000	0.0002	0.2791	0.1649	0.3601
Runs test	26.9561	-17.2798	11.2888	-9.1257	6.2382	0.2010
Runs test p-val	0.0000	0.0000	0.0000	0.0000	0.0000	0.8407
Logistic map r=3.5 with N(0,0.1) noise						
Scale factor	**1**	**2**	**3**	**4**	**5**	**10**
SampEn(2)	1.4431	2.2351	1.8983	2.3735	2.2532	2.2824
PE(5)	0.6781	0.9561	0.7941	0.9449	0.8389	0.8578
Perm test(5)	936.3438	103.9875	246.6824	103.5751	169.9490	123.9256
Perm test(5) p-val	0.0000	0.8349	0.0000	0.8421	0.0015	0.3601
Runs test	28.2849	2.1488	10.3024	-1.3942	4.2533	-0.4020
Runs test p-val	0.0000	0.0317	0.0000	0.1633	0.0000	0.6877

The MSE analysis in table 3 shows a general increase in each metric as the scale factor increases. This is to be expected as the logistic map is originally deterministic. The downsampling procedure breaks the correlation between successive points, leading to a loss of regularity. As the logistic map with r=3.5 is periodic with period 4, the results for even scale factors (2,4) can be seen to differ greatly from the results for odd scale factors (1,3,5). If the results are separated into even and odd categories, it is again evident that increasing the scale factor leads to a decrease in the regularity of the series.

3.3 Santa Fe Time Series Competition – Set A

A univariate time series was obtained from set A of the Santa Fe Time series competition [9]. The time series consists of 1000 intensity measurements from a laser in a physics experiment that varies from periodic to chaotic pulsations. Additional time series were generated by adding a Gaussian noise component with mean 0 and standard deviation equal to the standard deviation of the original laser intensity series multiplied by 0.1, 0.2 and 1. Plots of the four time series are shown in Figure 6 below.

Figure 6 : Santa Fe time series competition – Set A



The original series exhibits heteroskedasticity and has varying regularity. Despite this, all four methods used were able to detect the increase in chaos with the addition of noise of increasing variance.

Figure 7 : Comparison for Santa Fe data

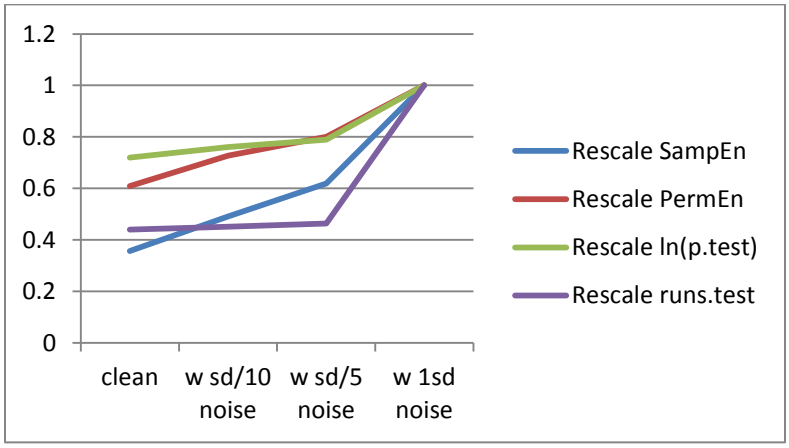


Table 3 : Scores for Santa Fe data

	clean	sd/10 noise	sd/5 noise	1 sd noise
SampEn(m=2)	0.7570	1.0441	1.3147	2.1233
PermEn(5)	0.5809	0.6933	0.7631	0.9529
Perm. test(5)	1562.7060	1045.5370	817.5509	198.3881
Perm. test(5) p-val	0.0000	0.0000	0.0000	0.0000
Runs test	-15.3711	-14.9967	-14.6170	-6.7707
Runs test p-val	0.0000	0.0000	0.0000	0.0000

The test statistics for all four measures show increasing chaos in the time series. However, the entropy statistic returned by SampEn and PermEn for the series with 1 standard deviation noise are so high that it would be difficult to distinguish between the noisy data and a purely random series. On the other hand, both the permutation test and the runs test have negligible p-values which would lead to rejection of the null hypothesis that the data originated from a random process.

MSE analysis was performed on the original clean series as well as the series with 0.2*standard deviation noise, with the results shown in Figure 8. As the original data is clearly periodic, it is again expected that the downsampling procedure will result in increasing scores as the scale factor increases, due to the reduction in correlations between successive points.

Figure 8

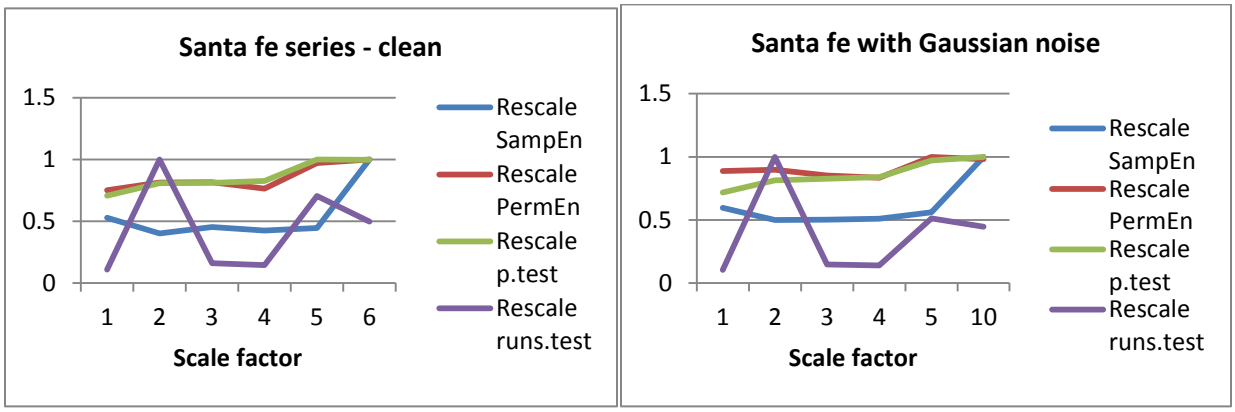


Table 4 : MSE analysis for Santa Fe data

Santa fe - clean						
Scale factor	**1**	**2**	**3**	**4**	**5**	**10**
SampEn(2)	0.7570	0.5752	0.6507	0.6111	0.6406	1.4328
PE(5)	0.5809	0.6306	0.6316	0.5927	0.7528	0.7734
Perm test(5)	1562.7060	622.3253	610.2527	540.2703	181.9454	183.8897
Perm test(5) p-val	0.0000	0.0000	0.0000	0.0000	0.0002	0.0001
Runs test	-15.3711	1.7011	10.5226	11.6606	2.4102	3.4173
Runs test p-val	0.0000	0.0889	0.0000	0.0000	0.0159	0.0006
Santa fe with N(0,0.2*sd) noise						
Scale factor	**1**	**2**	**3**	**4**	**5**	**10**
SampEn(2)	1.3401	1.1230	1.1285	1.1460	1.2615	2.2513
PE(5)	0.7443	0.7534	0.7135	0.6995	0.8386	0.8236
Perm test(5)	939.9436	415.9501	381.2034	348.3164	157.9526	135.9184
Perm test(5) p-val	0.0000	0.0000	0.0000	0.0000	0.0098	0.1376
Runs test	-14.8702	1.5221	10.3024	10.9002	2.9773	3.4173
Runs test p-val	0.0000	0.1280	0.0000	0.0000	0.0029	0.0006

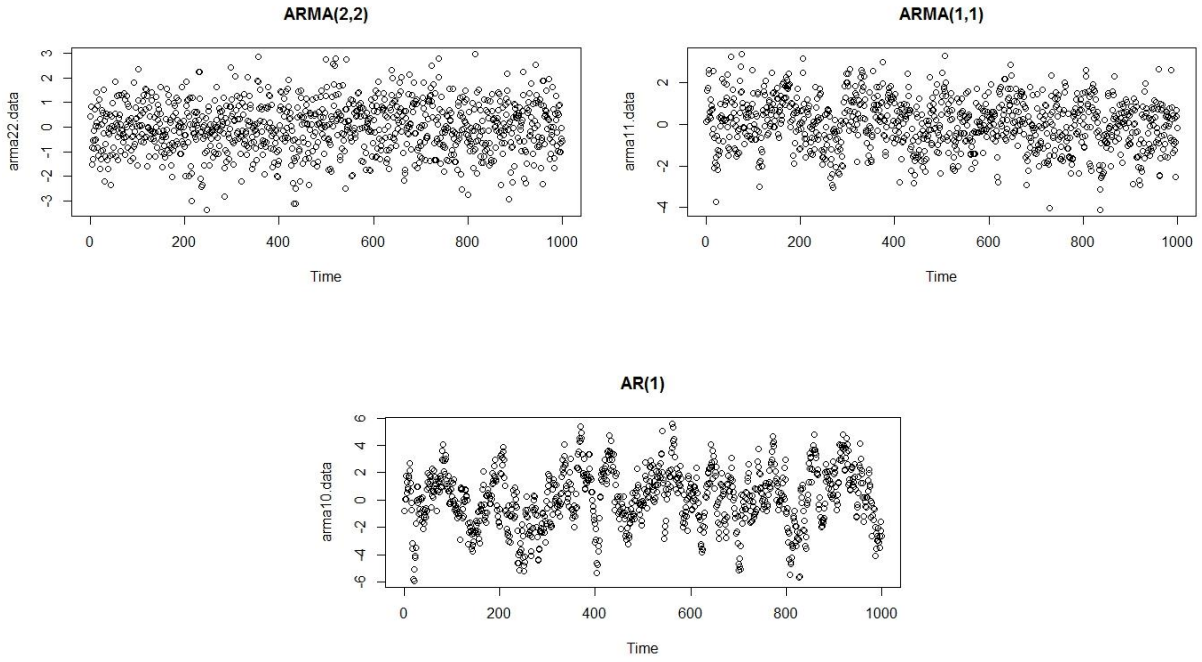
Table 4 shows that the scores generally increase as the scale factor increases, indicating the detection of a loss in regularity caused by the downsampling process. While the runs test statistic again displays a large variation as the scale factor is varied, it has the advantage of returning a very low p-value even

with a scale factor of 10, whereas the other 3 metrics return scores that are similar to those from a random series.

3.4 ARMA processes

The ARMA data set consists of 3 time series obtained by simulating several ARMA processes. The first series comes from an ARMA(2,2) process with AR coefficients of (0.9, -0.2) and MA coefficients of (-0.7, 0.1). The second series is an ARMA(1,1) process with an AR coefficient of 0.7 and MA coefficient of -0.2. The third and last series is an AR(1) process with coefficient 0.9. 1000 points were generated from each series, the plots which are shown below in Figure 8.

Figure 9 : Plots of varying ARMA processes



From inspection of the plots, increasing regularity can be seen when comparing the time series from the ARMA(2,2) process to the ARMA(1,1) process to the AR(1) process. Thus, the measures should show decreasing rescaled scores, as validated in Figure 10.

Figure 10 : Comparison for ARMA processes

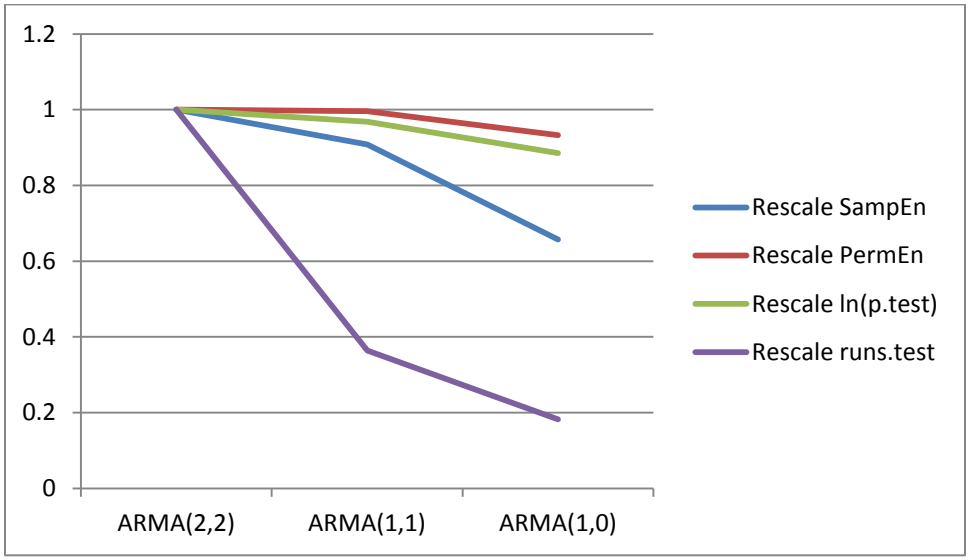


Table 4: Score for ARMA data

	ARMA(2,2)	**ARMA(1,1)**	**ARMA(1,0)**
SampEn(m=2)	2.2286	2.0238	1.4650
PermEn(5)	0.9833	0.9795	0.9173
Perm. test(5)	126.3924	147.9911	236.7858
Perm. test(5) p-val	0.3041	0.0369	0.0000
Runs test	-3.9865	-10.9470	-21.8939
Runs test p-val	0.0001	0.0000	0.0000

The test statistics for all four measures indicate a decreasing measure of randomness or complexity. SampEn, PermEn and the permutation test are unable to distinguish between an ARMA(2,2) process and a purely random process, while the runs test is able to do so with a very small p-value. SampEn and PermEn still have problems with an ARMA(1,1) process, while the permutation test is able to reject the null hypothesis of a random process at a 0.05 significance level.

The MSE analysis of all 3 processes is shown in Figures 11a – 11c. Both the permutation test statistic and permutation entropy show minimal change as the scale factor increases, whereas there is significant variation in the sample entropy score and the runs test statistic.

Figure 11a

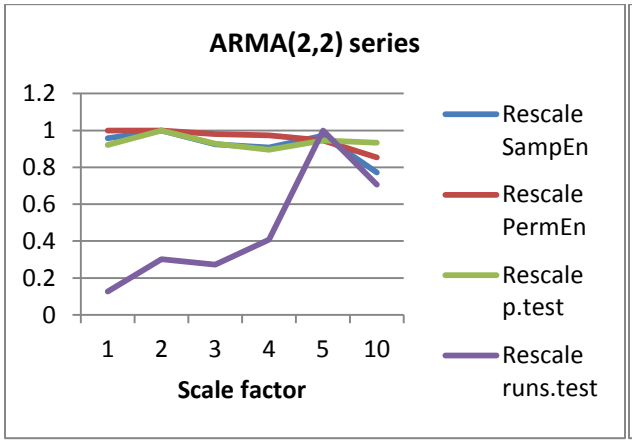


Figure 11b

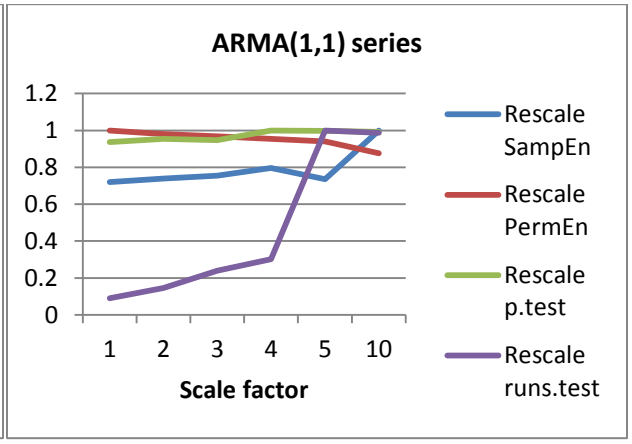


Figure 11c

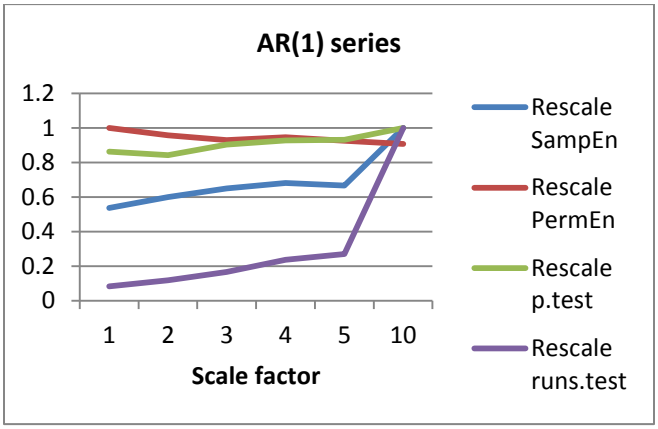


Table 5 : MSE analysis for ARMA models

ARMA(2,2)						
Scale factor	**1**	**2**	**3**	**4**	**5**	**10**
SampEn(2)	2.1533	2.2500	2.0808	2.0424	2.1864	1.7383
PE(5)	0.9846	0.9758	0.9559	0.9497	0.9235	0.8329
Perm test(5)	131.1921	89.5892	126.7042	151.5636	115.9652	123.9256
Perm test(5) p-val	0.2096	0.9797	0.2974	0.0235	0.5616	0.3601

Runs test	-4.4927	-1.8802	-2.0824	-1.3942	0.5671	0.8041
Runs test p-val	0.0000	0.0601	0.0373	0.1633	0.5706	0.4214
ARMA(1,1)						
Scale factor	**1**	**2**	**3**	**4**	**5**	**10**
SampEn(2)	2.0238	2.0806	2.1228	2.2407	2.0680	2.8134
PE(5)	0.9795	0.9608	0.9486	0.9348	0.9217	0.8586
Perm test(5)	147.9911	135.1838	141.2470	108.3740	109.9670	111.9328
Perm test(5) p-val	0.0369	0.1474	0.0802	0.7476	0.7114	0.6642
Runs test	-10.9470	-6.8045	-4.1648	-3.2954	-0.9924	-1.0051
Runs test p-val	0.0000	0.0000	0.0000	0.0010	0.3210	0.3149
AR(1)						
Scale factor	**1**	**2**	**3**	**4**	**5**	**10**
SampEn(2)	1.4650	1.6371	1.7735	1.8615	1.8197	2.7300
PE(5)	0.9173	0.8782	0.8527	0.8691	0.8503	0.8322
Perm test(5)	236.7858	269.5677	184.8755	161.1613	157.9526	111.9328
Perm test(5) p-val	0.0000	0.0000	0.0001	0.0061	0.0098	0.6642
Runs test	-21.8939	-15.2205	-10.8504	-7.6048	-6.6636	-1.8092
Runs test p-val	0.0000	0.0000	0.0000	0.0000	0.0000	0.0704

Examination of table 5 reveals that similar to the case with random data, the permutation entropy score decreases almost monotonically as scale factor increases. Sample entropy and the permutation test statistic do not display any obvious trend with the change in scale factor.

3.5 Congestive Heart Failure – Normal Sinus Rhythm data

Entropy measures are commonly used in the analysis of physiologic time series. For the final evaluation of entropy measures against tests for randomness, two separate inter-beat (RR) interval time series were evaluated[1]. The first series is comprised of measurements from 44 patients with congestive heart failure (CHF)[10], while the second series is comprised of measurements from 54 patients with normal sinus rhythm (NSR). The series length varies from 75,546 data points to 147,880 data points for the CHF group, and from 76,927 data points to 136,528 data points for the NSR group.

[1] Provided by Douglas E. Lake, UVA Department of Medicine

SampEn, PermEn, the permutation test and the runs test were applied to each of the time series in both groups. Similar to the previous experiments, for SampEn the parameters m=2 and r=0.2 was chosen, while for PermEn and the permutation test the tuple size was set to 5. Figures 12a-d show the results of the various measures on CHF patients (red box plot) and NSR patients (blue box plot).

Figure 12a : SampEn

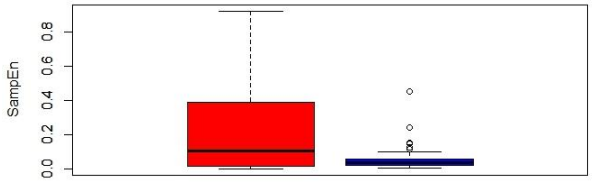


Figure 12b : PermEn

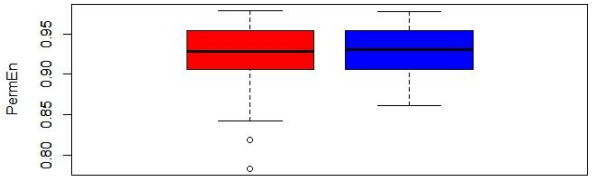


Figure 12c : Permutation test

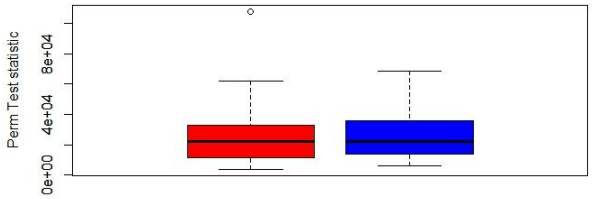
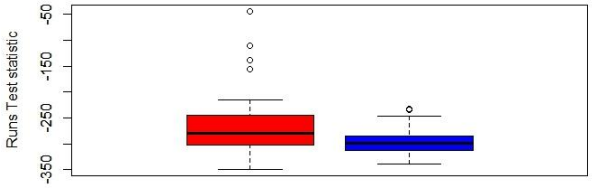


Figure 12d : Runs test



It is readily apparent that SampEn scores are markedly different between the two groups, with a lower mean score and much lower variance. This suggests that the NSR patients have more regular inter-beat interval times. The scores for PermEn and the permutation test are not markedly different between the two groups, again highlighting the similarity between these two measures. Lastly, the runs test statistic provides similar results as the SampEn score, except to a lesser degree. The absolute value of the runs test statistic for NSR patients is larger, again suggesting that NSR patients have inter-beat interval times that are more regular. The variance of the runs test statistic is also lower, similar to what was found for SampEn.

A two-sample t-test of the scores from both groups quantifies the difference shown in the box plots. For both SampEn and the runs test, we are able to reject the null hypothesis that the two samples have the same mean at a 0.05 significance level, whereas for PermEn and the permutation test we are unable to reject the null hypothesis at a 0.05 significance level.

4. Discussion and future work

Entropy based methods such as sample entropy and permutation entropy are able to quantify the degree of regularity present in a series and utilize this measure as a means of comparing the complexity of different time series. In a similar fashion, established tests for randomness such as the permutation test and the runs test examine a series for the presence of underlying structure in order to determine the likelihood of the series originating from a random distribution.

Experimental analysis shows that the test statistics of the permutation test and the runs test vary in a fashion that is highly correlated to SampEn and PermEn scores. Thus, these tests may be able to provide similar information regarding the complexity of time series when comparing multiple data sets. Furthermore, such statistical tests also have the advantage of having well-known statistical distributions which can provide probabilistic information on the likelihood of the originating process being random. In comparison, a measure of entropy from SampEn and PermEn by itself may not provide enough information to make the aforementioned distinction, even with the application of the MSE framework. In some cases, the p-values from the permutation test and the runs test can provide additional information that is not detectable by changes in SampEn and PermEn.

Further study should be carried out on the potential applications of various other tests for randomness in conjunction with entropy-based measures to gain further insight into the complexity of time series, which may provide additional predictive power.

References

[1] Shannon, Claude E. (July/October 1948). "A Mathematical Theory of Communication". Bell System Technical Journal 27 (3): 379–423.

[2] Steven M. Pincus, Approximate entropy as a measure of system complexity, *Proc Natl Acad Sci USA* 88: 2297–2301, 1991.

[3] Richman JS, Moorman JR, Physiological time-series analysis using approximate entropy and sample entropy, Am J Physiol Heart Circ Physiol. 2000

[4] Christoph Bandt, Bernd Pomp, Permutation Entropy: A Natural Complexity Measure for Time Series, Phys. Rev. Lett. 88, 174102 (2002)

[5] Costa M, Goldberger AL, Peng CK, Multiscale entropy analysis of complex physiologic time series, Phys Rev Lett. 2002

[6] Donald E. Knuth, The Art of Computer Programming, Volume 2: Seminumerical Algorithms. Third Edition (Reading, Massachusetts: Addison-Wesley, 1997)

[7] Douglas E. Lake, Joshua S. Richman, M. Pamela Griffin and J. Randall Moorman, Sample entropy analysis of neonatal heart rate variability, Am J Physiol Regul Integr Comp Physiol, 283:R789-R797, 2002

[8] http://cran.r-project.org/web/packages/lawstat/index.html

[9] http://www-psych.stanford.edu/~andreas/Time-Series/SantaFe.html

[10] http://www.physionet.org/physiobank/database/nsr2db/

Chapter 3 : Developing an ICU scoring system with interaction terms using a genetic algorithm

Abstract

ICU mortality scoring systems attempt to predict patient mortality using predictive models with various clinical predictors. Examples of such systems are APACHE, SAPS and MPM. However, most such scoring systems do not actively look for and include interaction terms, despite physicians intuitively taking such interactions into account when making a diagnosis. One barrier to including such terms in predictive models is the difficulty of using most variable selection methods in high-dimensional datasets.

A genetic algorithm framework for variable selection with logistic regression models is used to search for significant two-way interaction terms in a clinical dataset of adult ICU patients. The dataset is also split according to category of diagnosis upon admittance to the ICU, and separate models are built for each category. The models had good discrimination across all categories, with a weighted average AUC of 0.84 (and > 0.90 for several categories) and the genetic algorithm was able to find many significant interaction terms, some of which may be able to provide greater insight into mortality prediction for health practitioners. The GA selected models had improved performance against stepwise selection and random forest models, and provides greater flexibility in terms of variable selection by being able to optimize over any modeler-defined model performance metric instead of a specific variable importance metric.

Keywords: ICU scoring system, predictive modelling, genetic algorithm, variable selection, interaction terms

3.1 Introduction

Predictive modelling in healthcare is a rapidly growing field. Recent innovations in information systems use in hospitals has resulted in a massive increase in the availability and accuracy of patient electronic health records (EHRs) and other sources of medical data. This Data Science boom has enabled the development of more predictive analytics tools to aid health

practitioners in tasks such as diagnosing illnesses, assessing the likelihood of patient readmission, and predicting patient mortality.

Many predictive scoring systems for adult ICU patient mortality have been developed. Among the most popular are the Acute Physiology and Chronic Health Evaluation (APACHE) score, the Mortality Probability Models (MPM) and the Simplified Acute Physiology Score (SAPS). Most of these predictive models are built using physiological, clinical or therapeutic variables that are routinely collected in the ICU, either as a first day snapshot or dynamically updated throughout a patient's ICU stay. Furthermore, most such scoring systems are based on a form of logistic regression to predict a patient's probability of mortality.

The Acute Physiology And Chronic Health Evaluation (APACHE) score was developed by Knaus et. al. [1] to assess the severity of illness of critically ill adult patients admitted to the intensive care unit (ICU). The first APACHE model consisted of 34 physiologic predictors selected using expert judgement. Further refinements to the APACHE model have followed with APACHE II, III and IV. APACHE II has been widely used in many hospitals and healthcare facilities for benchmarking purposes [2].

The APACHE II score is based on several clinical and physiologic measurements taken when a patient is first admitted to the ICU [3]. For APACHE II, the score is calculated from the following 13 predictors : age, alveolar-arterial gradient (A-aO_2) or partial pressure arterial oxygen (PaO_2) depending on the fraction of inspired oxygen (FiO_2),rectal temperature, mean arterial pressure (MAP), arterial pH, heart rate, respiratory rate, sodium (serum), potassium (serum), creatinine, hematocrit, white blood cell count and Glasgow Coma Scale score. First, each predictor value is mapped to a numeric integer score according to where the value falls in various pre-determined ranges of possible values. For example, the score for age is determined according to the following function:

$$score\_age = \begin{cases} 0\ if\ 0 \leq age \leq 44 \\ 2\ if\ 45 \leq age \leq 54 \\ 3\ if\ 55 \leq age \leq 64 \\ 5\ if\ 65 \leq age \leq 74 \\ 6\ if\ 75 \leq age \leq 130 \end{cases}$$

After each predictor is mapped accordingly, the sum of all predictor scores is used in a logistic regression model to predict mortality. APACHE III expanded on APACHE II by including five additional physiologic predictors to the APS component, and included three two-way interaction terms as well [4]. The latest version, APACHE IV, uses a multivariate logistic regression model with a much larger dataset (110,588 patients) compared to its predecessors [5].

The Simplified Acute Physiology Score (SAPS) was originally based on the APS predictors included in APACHE [6]. Expert judgement was used to reduce the number of predictors to 13 physiologic variables and patient age. SAPS II later included 4 additional demographic variables, bringing the total up to 17 (12 physiologic, 5 demographic) [7]. The predictors were assigned a score depending on the range (similar to APACHE), with the sum of scores then being used in a logistic regression for patient mortality.

The Mortality Probability Model (MPM) scoring system was developed using 12 variables in a multi-variate logistic regression model [8]. Initially based only on data at the time of admission, further studies incorporated data taken 24 hours and 48 hours after admission. MPM II was later developed which included models built for data at admission, after 24 hours, after 48 hours and after 72 hours [9]. Two-way interaction terms were considered in MPM II, but were eventually rejected for not satisfying the author's criteria for inclusion.

The aforementioned ICU scoring systems have been validated with good performance in numerous studies [10]. However, a common point among these scoring systems is that they are mainly developed using predictors selected by subject matter expert judgement and mostly do not include interaction terms (APACHE III and possibly APACHE IV are exceptions). However, intuitively when predicting patient mortality, it is likely that the existence of certain conditions in conjunction may pose a much greater health risk than when these conditions exist independently. Many physicians would naturally take into account the interplay of all physiologic variables when making a diagnosis, instead of considering each variable independently. In a complex problem such as predicting mortality, there may be many interaction effects that can give additional power to the model. In many cases, health

practitioners are aware of such effects based on their experience and judgement but have no way of quantifying the strength of the interactions due to the lack of research into the inclusion of interaction terms. Thus, for the sake of model parsimony interaction terms are often omitted (e.g. the MPM II model rejected interaction terms if there was no "clinical plausibility" behind them [9]). However, it is also possible that beneficial interaction effects exist which are currently unknown to health practitioners and therefore would not be included in a model designed mainly using expert knowledge.

Thus, this exploratory study aims to develop a prototype ICU mortality scoring system using machine learning methods (a genetic algorithm) for variable selection instead of relying solely on expert knowledge. By evaluating the efficacy of models with interaction terms included, we aim to explore the potential benefits of using a variable selection method that can handle a large number of interactions to develop such models and hopefully find novel interactions that may not be well-known to health practitioners.

3.2 ICU mortality dataset

For this study, we obtained a dataset of 224,418 patient records with 12 binary comorbidities, 5 categorical clinical predictors, and 2 numeric predictors[2]. A similar dataset was used to evaluate APACHE IV against APACHE III [5]. Table 1 below summarizes the list of predictors included in the dataset.

[2] Private communication with Dr. Andrew Kramer, formerly of the Cerner Corporation.

Table 3.2.1 : Predictors in ICU dataset

Binary predictors	**Categorical predictors**	**Numeric predictors**
operative, emergency, aids, myeloma, lymphoma, cirrhosis, tumorwm, immunosup, hepfail, copd, diabetic, dialysis	visit priorloc gender ethnic dx_group	age APS

The first two binary predictors represent whether a patient is in the ICU for an operative or emergency procedure. The remaining binary predictors represent the absence or presence (0 or 1 respectively) of the listed comorbidities in the patient upon being admitted to the ICU. For the numeric predictors, “age” lists the patient’s age in years (integer) while the Acute Physiology Score (APS) is an integer score based on a regression model using 12 clinical predictors, some of which are included in the list of APACHE II predictors.

The first categorical predictor, “visit”, indicates how many times the patient has been admitted to the ICU and ranges from 1 to 9. “Priorloc” indicates the patient’s location prior to entering the ICU, e.g. home, other hospital ICU etc. “Gender” and “ethnic” indicate the sex and ethnicity (6 levels) of the patient respectively. “Dx_group” stores the patient’s diagnosis code, which is given by a physician upon admittance to the ICU. The diagnosis code is assigned based on the physician’s diagnosis of the patient’s condition. The diagnosis code is a factor (with 122 levels in this dataset) which can be grouped into 16 categories. Note that a patient being admitted to the ICU can exhibit multiple conditions, e.g. head trauma and intercerebral hemorrhage. However, only a single primary condition (as judged by the attending physician) is recorded in the data. Thus, each patient can only be associated with a single “dx_group” value.

Unfortunately, while the ICU dataset was closely related to the data used to develop the APACHE models, several key variables were omitted (Glasgow Coma Scale, AaDO2/PaO2, pH arterial, potassium). Thus, we were unable to calculate the APACHE II score (or any of the other commonly used ICU scoring systems) for the patients in the dataset as a baseline comparison.

3.3 Data preprocessing

The dataset included two binary predictands, “icudead” and “hosdead”. These labels represent whether the patient passed away in the ICU or subsequently in the hospital after being discharged from the ICU. For the purposes of this study, we only considered patient mortality in the ICU as there could be multiple complicating factors involved in hospital mortality that are not captured in the dataset. Thus, all patients who passed away in the hospital were removed, leaving only patients that survived or passed away in the ICU. Records that contain missing data in the categorical/binary predictors were removed, while records with missing data in the numeric predictors were replaced by the mean. As a result, the final dataset used for the analysis consisted of 154,281 patient records.

Several issues arose during the initial analysis of the dataset. Firstly, the APS predictor is an aggregate measure of several clinical predictors. Thus, it provides a general indication of the patient’s health condition but does not provide information on the factors contributing to the score. While it performed adequately as an input to the original APACHE formulation, in order to explore potential interaction terms (especially with comorbidities) it would be more meaningful to expand the APS into its constituent components. Doing so added 12 additional numeric predictors to the dataset. These numeric predictors, together with the age of the patient, were scaled to have mean 0 and variance 1 in order to reduce the effects of multi-collinearity.

Secondly, the “visit” variable was changed from a 9 level factor to a 2 level factor indicating whether the patient was a first time visitor to the ICU, or a repeat visitor. Repeated visits to the ICU could be indicative of additional health complications or a poor health condition in general, leading to higher risk of mortality. However, the vast majority of patients had “visit” levels of

either 1 or 2 (>98%), with a small minority having more than 2 visits. As such, we could combine all “visit” levels 2 or greater into a single level, greatly reducing the complexity of the model while retaining most of the predictive power.

Lastly, the “dx_group” predictor with 122 levels resulted in a very sparse matrix with many diagnosis codes belonging to very few patients, or none at all. In addition, consultations with subject matter experts (physicians working in the University of Virginia Hospital ICU) revealed that in many cases the initial diagnosis is subjective and the diagnosis code assigned to the patient can vary substantially from physician to physician. Thus, the existing data on diagnosis codes is likely to be fairly noisy. However, there is less contention regarding the category of diagnosis. For example, it may be unclear whether a patient is suffering from bacterial pneumonia or viral pneumonia, but most physicians would categorize the diagnosis as a respiratory condition. Following this line of reasoning, the various diagnosis codes were aggregated into the following 12 categories :

Table 3.3.1 : Diagnosis categories in ICU dataset

Category	**# of patients**
Cardiovascular diagnosis	52,630
Cardiovascular surgery	9,690
Respiratory diagnosis	23,047
Respiratory surgery	3,478
Neurologic diagnosis	20,222
Neurologic surgery	6,510
Gastrointestinal diagnosis	11,422
Gastrointestinal surgery	8,975
Trauma diagnosis	6,869
Trauma surgery	2,261
Metabolic diagnosis	6,839
Genitourinary diagnosis	2,338
Total	***154,281***

Furthermore, our discussion with subject matter experts suggested that it would likely improve model performance to subset the data according to the categories shown above. A patient

admitted to the ICU for trauma injuries could have a very different set of mortality predictors than a patient admitted for respiratory problems. Many of the original ICU scoring systems were intentionally designed for ease of use with pen and paper calculations, and developing different models for different diagnosis codes would have greatly complicated the scoring process. However, with the widespread use of information technology in hospitals it should no longer be a requirement to be constrained to a single aggregated model for all patient conditions. By developing a model for each diagnosis category, we are also able to better explore potential interaction terms without the confounding effects of other conditions. Appendix 3A lists the final predictors considered in the models, while Appendix 3B shows the descriptive statistics for categorical predictors from each subset.

For each subset, logistic regression models were used to fit the data to predict ICU mortality. This choice was informed by several factors. Firstly, logistic regression models are widely used in the medical community and are well-understood by physicians, allowing for easier acceptance of the resulting models. Predictions from logistic regression models are also easier to calculate without special software and can be performed using spreadsheets or mobile apps, compared to models such as random forests or artificial neural networks. Secondly, many studies of ICU mortality have used logistic regression models with similar predictors and demonstrated good performance. One of the primary concerns with logistic regression models is the possible presence of non-linear predictors, which are common in medicine due to the prevalence of homeostatic processes in living organisms. However, empirical results show that logistic regression models perform well on many medical datasets even without first applying transformations to non-linear predictors. Lastly, logistic regression models are easily interpretable, especially with regards to interaction terms. Interaction terms are explicitly defined in logistic regression models and thus their effects can be more easily isolated and evaluated.

After deciding on the model, we now have to determine the appropriate quantitative metric to use for model evaluation. As the GA provides great flexibility in the choice of fitness function, there are many possible options. The area under the Receiver Operating Characteristic (ROC)

curve is a metric that is commonly used in machine learning for model comparison and has also seen widespread use and acceptance in the medical community. The ROC curve is derived by using the model's predictions to plot the true positive rate (TPR) against the false positive rate (FPR) for various values of the decision threshold. The area under the ROC curve (AUC) can therefore be used as a metric of a model's discriminative power, with a larger AUC indicating that a model has a higher probability of ranking a randomly chosen positive instance higher than a randomly chosen negative instance. It should be noted that the AUC alone should not be taken as a definitive measure of a model's effectiveness. A model with a higher AUC does not necessarily perform better than another model with a lower AUC, as the AUC represents the models' performance across all possible thresholds. When a model is used for classification a specific threshold has to be chosen in order for a class prediction to be made, and the relative performance of the models at that specific threshold could well differ from their AUC rankings. Nevertheless, we chose to use AUC as the model evaluation criteria as the AUC serves well as a general indicator of model performance and has been used extensively in evaluating APACHE, SAPS and MPM.

For each diagnosis subset of the ICU dataset, the GA framework was used to perform variable selection for a logistic regression model, using AUC as the fitness function. Each subset was split into ten folds for cross-validation using random sampling without replacement, with the size of folds 1-9 being set to floor($\frac{N}{10}$), where N is the total number of records in the subset. Fold 10 contains the remaining records after folds 1-9 have been drawn. Each candidate solution consists of a set of predictors, which are then evaluated on each test fold in turn after being trained on the remaining nine folds. The ten resulting AUC scores are then averaged to obtain the overall AUC score. For all subsets, the GA's parameters were set to the following:

Table 3.3.1 : GA parameter settings

Population size	30
Min/Max number of predictors	5/100
Maximum number of generations	250
Recombination probability	0.5 to 0.2
Mutation probability	0.01 to 0.2

For a detailed account of the GA procedure, please refer to Chapter 1. For each diagnosis subset, 5 runs were performed using different initial random number generator seeds, and the best performing GA result was chosen. The following section describes the results from each subset, as well as provides some comparisons with other modelling methods.

3.4 Results

Due to the limitations of the variables provided in the dataset, we were unable to compare the AUC of the GA selected model against other ICU scoring systems like APACHE, SAPS II and MPM II. To provide a comparison, for each subset we developed a logistic regression model using stepwise selection according to AIC and a random forest model with 500 trees. The same ten folds used for the GA selection process were used to evaluate the AUC with each of the stepwise selected logistic regression models and random forest models. Table 3.4.1 below shows the mean AUCs obtained for the stepwise-selected logistic regression model, the random forest model, and the GA-selected logistic regression model respectively. For each subset, the standardized mortality ratio (SMR) of the GA-selected logistic regression models was not significantly different from 1.0, indicating no major differences between the observed number of deaths and the expected number of deaths.

Table 3.4.1 : Mean AUC for logistic regression (stepwise), random forest and logistic regression (GA)

	Stepwise	Random forest	GA	GA vs Step	GA vs RF
	AUC	***AUC***	***AUC***	***p-vals***	***p-vals***
Cardiovascular diagnosis	0.8187	0.8605	0.8300	0.003906	0.001953
Cardiovascular surgery	0.8614	0.8684	0.8921	0.01367	0.08398
Respiratory diagnosis	0.7719	0.7761	0.7852	0.001953	0.1602
Respiratory surgery	0.8290	0.8213	0.9159	0.009766	0.005859
Neurologic diagnosis	0.7824	0.8390	0.8050	0.01367	0.001953
Neurologic surgery	0.8833	0.8678	0.9200	0.001953	0.001953
Gastrointestinal diagnosis	0.8265	0.8383	0.8426	0.003906	0.4922
Gastrointestinal surgery	0.8199	0.8545	0.8692	0.01367	0.001953
Trauma diagnosis	0.8170	0.8805	0.8597	0.003906	0.08398
Trauma surgery	0.8383	0.8896	0.9065	0.003906	0.1934
Metabolic diagnosis	0.8560	0.8580	0.8952	0.001953	0.009766
Genitourinary diagnosis	0.7844	0.7855	0.8599	0.001953	0.02734

It can be seen that the discrimination of the GA-selected model is fairly good, ranging from 0.7852 to 0.9200 across the various subsets. The GA-selected model significantly outperformed the stepwise selected model in all 12 categories (at a 0.05 significance level), while the random forest model was better in 2 categories and worse in 5 categories. In particular, the GA-selected model performed markedly better than both stepwise selection and random forest in the "respiratory surgery" and "genitourinary diagnosis" categories.

Tables 3.4.2 and 3.4.3 below summarize the significant predictors in each category. Each column represents a diagnosis category, while the rows represent the main effects terms. A highlighted cell in a column indicates that the main effect is included in the model for the indicated diagnosis category. The numbers in each cell indicate the variables with which the term has significant pair-wise interactions. For example, in the model for cardiovascular diagnosis the "visit" term is a significant main effect with no interaction terms, while the "age" term has significant interactions with "dialysis", "temp", "sodium" and "album".

Table 3.4.2 : Model summary for diagnosis categories 1-6

		Cardio diag	Cardio surg	Resp diag	Resp surg	Neuro diag	Neuro surg
1	**visit**		4,20	20,28			26
2	**ipriorloc**	12,18		11,14		4,16,21	
3	**gender**						23
4	**age**	16,18,25,28	1,26	11		2,22,24	
5	**operative**						
6	**emerg**		29				19
7	**aids**						
8	**myeloma**	25				12,23	
9	**lymphoma**						
10	**cirrhosis**					24	
11	**tumorwm**		18	2,4,12			
12	**imm.sup**	2		11		8,28	
13	**hepfail**	26,27		28		25	
14	**copd**	19,22		2,26		22	
15	**diabetic**	22,27	16		21	25,27	22
16	**dialysis**	4,26	15,21,26	26		2	
17	**ethnic**	18,25	24			19	
18	**temp**	2,4,19,20,27	11,28	19	20	20,22,26,28	19,22, 29
19	**map**	20,21,22,25, 27	20	18,20,21,23, 26		17,21,28,29	6,18,23
20	**hr**	23	1,19,21	1,19	18,21	18	
21	**rr**	25	29	19	15,20,22,25, 27	2,19,22,23,27	
22	**urine**	27			23	4,14,18,21,23,24, 25	15,18
23	**wbc**	26,28		29	22	8,21,22,24	3,19
24	**hcrit**	26	17			4,10,22,23	25
25	**sodium**	4,8,17,19				13,15,22	
26	**creat**	13,16,23,24	4,16	14,16,19		18,27	1
27	**gluc**	13,15,18,19, 22				15,21,26,28	
28	**album**	4,23	18	13,29		12,18,19,27	
29	**bili**		6,21	23,28			18

Table 3.4.3 : Model summary for diagnosis categories 7-12

		Gastro diag	Gastro surg	Trauma diag	Trauma surg	Meta diag	Genito diag
1	visit	15				24	
2	ipriorloc	22,25	13	28			
3	gender		18	18,23	26	28	22
4	age	22		22,25	19,21,27	11	
5	operative						
6	emerg				22		
7	aids						
8	myeloma						
9	lymphoma						
10	cirrhosis						15
11	tumorwm	22		22,24,27		4	
12	imm.sup	23					27
13	hepfail	28,29	2,23				
14	copd			15,23	26,28		
15	diabetic	1,26		14			10
16	dialysis	24,25,26	22	20	20		27,28,29
17	ethnic			18			
18	temp		3,22,24	3,17,19,21,23	26	24,26	
19	map	20	23	18,23,25	4,23,26	20,22,28	20
20	hr	19,23		16,21,25,27	16,25,29	19	19
21	rr			18,20,25,29	4,25		22,28
22	urine	2,4,11	16,18	4,11	6	19,27	3,21
23	wbc	12,20	13,19	3,14,18,19,28	19	25,26	
24	hcrit	16	18	11		1,18	
25	sodium	2,16,27		4,19,20,21	20,21	23	
26	creat	15,16,29			3,14,18,19	18,23	
27	gluc	25		11,20,29	4	22	12,16
28	album	13		2,23	14	3,19	16,21
29	bili	13,26		21,27	20		16

The current implementation of the GA does not directly select for model sparsity in the fitness function, which results in the GA-selected models having a fairly large number of predictors. While a penalty for model size could be added to the fitness function, doing so comes with significant downsides. Firstly, such a penalty function could interfere with the GA selection

process by forcing the GA to become too greedy and prematurely weed out predictors that may initially provide little improvement to the AUC, but would improve the fitness in the presence of certain other predictors. Secondly, the determination of the appropriate penalty function is non-trivial and has a significant effect on the GA's performance. However, it should be noted that the larger size does not necessarily translate to a larger burden on data collection, as the majority of the additional variables are interaction terms derived from the original 29 main effects variables which are already routinely collected. Furthermore, the GA-selected models can be further refined using expert judgement or other variable selection methods, both of which become more viable once the number of potential predictors has been reduced using the GA.

As expected, the models for each diagnosis category differ substantially. However, the patient's age, number of visits and various APS predictors generally are significant in almost every category, which is consistent with the findings of other ICU scoring systems. The presence of diabetes and whether the patient is on dialysis are also significant in several models, while the presence of AIDS, myeloma, cirrhosis, and whether the patient was admitted for operative purposes is relatively insignificant. Further examination of the models also reveals some interesting observations. Firstly, the "emergency" predictor is significant in most of the diagnosis categories pertaining to surgery. Secondly, most of the models include significant interactions amongst the APS predictors (together with significant APS main effects terms). These interaction effects have not been included in other ICU scoring systems that use the APS score as an aggregate predictor. Thirdly, the GA was able to identify several interactions in the dataset that could potentially be avenues for further study. For example, the ethnicity of the patient is significant in several categories (cardiovascular diagnosis, neurological diagnosis, trauma diagnosis) along with interactions with APS predictors such as sodium and temperature. The gender of the patient also has significant interactions with APS predictors in various diagnosis categories.

3.5 Discussion and future work

The results of the study show that there is potential benefit in utilizing machine learning methods, in this case a genetic algorithm for variable selection, in developing ICU scoring systems which include interaction terms. Using AUC as a measure of model performance, the GA-selected logistic regression models had comparable or better discrimination than stepwise-selected logistic regression models or random forest models. We also show that developing different models for various diagnosis categories rather than using a single model for all ICU patients may yield improved model performance as well as provide insight in the form of significant interaction terms for each particular diagnosis category. Thus, the GA selection process can serve as a useful first step in developing models to support physicians in predicting patient mortality.

However, the GA-selection procedure also comes with some notable drawbacks. The first is the procedure run-time, which can be very significant compared to other variable selection methods. On the other hand, the GA selection procedure is able to deal with an arbitrarily large number of potential predictors, unlike several common variable selection procedures. Furthermore, the long run-time is only applicable during model development (or updating) not during patient classification.

Secondly, there is no theoretical guarantee that the GA will find globally optimum models that generalize well. The models returned by the GA should be validated on another dataset that should ideally contain the same predictors used in other ICU scoring systems, which would allow a better comparison of the models with interaction effects. The GA could also be coupled with other variable selection procedures to try to prune the final number of predictors, which could make the models more generalizable to other datasets.

References

[1] Knaus WA, Zimmerman JE, Wagner DP, Draper EA and Lawrence DE. APACHE acute physiology and chronic health evaluation: a physiologically based classification system. Crit Care Med 1981;**9**(8):591-7.

[2] Nouira S, Belghith M, Elatrous S, Jaafoura M, Ellouzi M, Boujdaria R, et al. Predictive value of severity scoring systems: comparison of four models in Tunisian adult intensive care units. Crit Care Med 1998;**26**(5):852-9

[3] Knaus WA, Draper EA, Wagner DP and Zimmerman. APACHE II: A severity of disease classification system. Crit Care Med 1985;**13**:818-830.

[4] Knaus WA, Wagner DP, Draper EA, et al. The APACHE III prognostic system. Risk prediction of hospital mortality for critically ill hospitalized adults. Chest 1991; 100:1619.

[5] Zimmerman JE, Kramer AA, McNair DS, Malila FM. Acute Physiology and Chronic Health Evaluation (APACHE) IV: hospital mortality assessment for today's critically ill patients. Crit Care Med 2006; 34:1297.

[6] Le Gall JR, Loirat P, Alperovitch A, Glaser P, Pranthil C, Mathieu D et al.. A simplified acute physiology score for ICU patients. Crit Care Med 1984;**12**:975-977.

[7] Le Gall JR, Lemeshow S and Saulnier F. A new Simplified Acute Physiology Score (SAPS II) based on a European/North American multicenter study. JAMA1993;**270**(24):2957-2963.

[8] Lemeshow S, Teres D, Pastides H, Avrunin JS and Steingrub JS. A method for predicting survival and mortality of ICU patients using objectively derived weights. Crit Care Med 1985;**13**(7):519-525.

[9] Lemeshow S, Teres D, Klar J, Avrunin JS, Gehlbach SH and Rapoport J. Mortality Probability Models (MPM II) based on an international cohort of intensive care unit patients. JAMA 1993;**270**(20):2478-2486

[10] Kuzniewicz MW, Vasilevskis EE, Lane R, et al. Variation in ICU risk-adjusted mortality: impact of methods of assessment and potential confounders. Chest 2008; 133:1319.

Appendix 3A : List of predictors and descriptions

Name	Description	Type
visit	# of times patient has been admitted to ICU	Factor (1-9)
ipriorloc	Prior location of patient	Factor (emergency department, other floor, home, ICU transfer, other hospital, other hospital ICU, other, SDU, telemetry)
gender	Male or female	Factor (0 = Male, 1 = Female)
age	Patient age in years	Numeric
operative	Procedure is operative	Binary
emerg	Procedure is emergency	Binary
Aids	Presence of Acquired Immune Deficiency Syndrome (AIDS)	Binary
myeloma	Presence of myeloma (cancer of plasma cells)	Binary
lymphoma	Presence of lymphoma (cancer of lymphatic system)	Binary
cirrhosis	Presence of cirrhosis	Binary
tumorwm	Presence of tumor with metastasis	Binary
immunosup	Presence of immunosuppressive disorder	Binary
hepfail	Presence of hepatic failure	Binary
copd	Presence of chronic obstructive pulmonary disease	Binary
diabetic	Presence ofdiabetes	Binary
dialysis	Patient is on dialysis	Binary
ethnic	Ethnicity of patient	Factor (other unknown, African American, Asian, Caucasian, Hispanic, Native American)
temp	Temperature	Numeric
map	Mean arterial pressure	Numeric
hr	Heart rate	Numeric
rr	Respiratory rate	Numeric
urine	Urine output	Numeric
wbc	White blood cell count	Numeric
hcrit	Hematocrit	Numeric
sodium	Sodium level	Numeric
creat	Creatinine level	Numeric
gluc	Glucose level	Numeric
album	Albumin level	Numeric
bili	Bilirubin level	Numeric

Appendix 3B : Descriptive statistics of diagnosis subsets

(*only categorical predictors as numeric predictors have been scaled to have mean 0 and variance 1*)

Cardiovascular diagnosis

```
class     VISIT     gender   aids     myeloma   lymphoma  tumorwm   immunosup
 0:13217   1:14121   0:8213   0:14950   0:14739   0:14840   0:14361   0:13804
 1: 1783   2:  879   1:6787   1:   50   1:  261   1:  160   1:  639   1: 1196

 hepfail   copd      diabetic DIALYSIS
 0:14829   0:12563   0:9802   0:13770
 1:  171   1: 2437   1:5198   1: 1230

              ethnic            IPRIORLOC
 Other.Unknown   :  376   ED         :6162
 African.American: 2790   OHOSP      :2714
 Asian           :  201   FLOOR.OTHER:2667
 Caucasian       :11205   SDU        :1426
 Hispanic        :  271   TELEMETRY  : 830
 Native.American :  157   OHOSPICU   : 637
                          (Other)    : 564
```

Cardiovascular surgery

```
class   VISIT   gender   emerg   aids    myeloma  lymphoma tumorwm  immunosup
 0:9421  1:9388  0:5879  0:8268   0:9682   0:9639   0:9648   0:9574   0:9470
 1: 269  2: 302  1:3811  1:1422   1:   8   1:  51   1:  42   1: 116   1: 220

 hepfail  copd     diabetic DIALYSIS
 0:9673   0:8157   0:6885   0:9285
 1:  17   1:1533   1:2805   1: 405

              ethnic        IPRIORLOC
 Other.Unknown   : 234   ICUTRANS:  50
 African.American: 765   OPROOM  :5952
 Asian           : 102   RR      :3688
 Caucasian       :8396
 Hispanic        : 161
 Native.American :  32
```

Respiratory diagnosis

```
class     VISIT     gender   aids     myeloma   lymphoma  tumorwm   immunosup
 0:13558   1:13271   0:7657   0:14933  0:14746   0:14836   0:14003   0:13183
 1: 1442   2: 1729   1:7343   1:   67  1:  254   1:  164   1:  997   1: 1817

 hepfail   copd      diabetic  DIALYSIS
 0:14848   0:11253   0:10183   0:14297
 1:  152   1: 3747   1: 4817   1:  703

              ethnic            IPRIORLOC
 African.American: 2681   ED         :5530
 Asian           :  170   FLOOR.OTHER:3877
 Caucasian       :11425   OHOSP      :2193
 Hispanic        :  216   SDU        :1415
 Native.American :  149   TELEMETRY  : 994
 Other.Unknown   :  359   OHOSPICU   : 494
```

```
                        (Other)    : 497
```

Respiratory surgery

```
class   VISIT    gender   emerg   aids    myeloma  lymphoma cirrhosis tumorwm
 0:3402  1:3285   0:2124   0:3159  0:3468  0:3443   0:3444   0:3444    0:3030
 1:  75  2: 192   1:1353   1: 318  1:   9  1:  34   1:  33   1:  33    1: 447

 immunosup hepfail  copd     diabetic DIALYSIS
 0:3110    0:3465   0:2827   0:2795   0:3426
 1: 367    1:  12   1: 650   1: 682   1:  51

              ethnic        IPRIORLOC
 Other.Unknown    :  82   ICUTRANS:  14
 African.American: 363   OPROOM  :1493
 Asian            :  49   RR      :1970
 Caucasian        :2901
 Hispanic         :  55
 Native.American  :  27
```

Neurologic diagnosis

```
class     VISIT     gender    aids      myeloma   lymphoma  cirrhosis tumorwm
 0:18872   1:19596   0:10391   0:20153  0:20106   0:20151   0:19891   0:19590
 1: 1350   2:  626   1: 9831   1:   69  1:  116   1:   71   1:  331   1:  632

 immunosup hepfail   copd      diabetic  DIALYSIS
 0:19438   0:20096   0:18338   0:16087   0:19758
 1:  784   1:  126   1: 1884   1: 4135   1:  464

              ethnic            IPRIORLOC
 African.American: 3387   ED         :11229
 Asian            :  361   OHOSP      : 4725
 Caucasian        :14916   FLOOR.OTHER: 2175
 Hispanic         :  560   SDU        :  890
 Native.American  :  225   OHOSPICU   :  413
 Other.Unknown    :  773   TELEMETRY  :  383
                           (Other)    :  407
```

Neurologic surgery

```
class   VISIT   gender   emerg   aids    myeloma  lymphoma tumorwm  immunosup
 0:6350  1:6148  0:3244  0:5454  0:6495  0:6458   0:6463   0:5900   0:6120
 1: 157  2: 359  1:3263  1:1053  1:  12  1:  49   1:  44   1: 607   1: 387

 hepfail  copd     diabetic DIALYSIS
 0:6492   0:6012   0:5340   0:6430
 1:  15   1: 495   1:1167   1:  77

              ethnic        IPRIORLOC
 African.American: 745   ICUTRANS:  20
 Asian            : 172   OPROOM  :2573
 Caucasian        :5120   RR      :3914
 Hispanic         : 151
 Native.American  :  66
 Other.Unknown    : 253
```

Gastrointestinal diagnosis

```
class     VISIT     gender  aids      myeloma   lymphoma  tumorwm   immunosup
 0:10653   1:10683   0:6441  0:11385   0:11287   0:11339   0:10813   0:10598
 1:  769   2:  739   1:4981  1:   37   1:  135   1:   83   1:  609   1:  824

 hepfail   copd     diabetic DIALYSIS
 0:10502   0:9909   0:8099   0:10779
 1:  920   1:1513   1:3323   1:  643

              ethnic           IPRIORLOC
 African.American:1883   ED          :5027
 Asian           : 188   FLOOR.OTHER:2494
 Caucasian       :8636   OHOSP       :2353
 Hispanic        : 253   SDU         : 547
 Native.American : 154   TELEMETRY   : 389
 Other.Unknown   : 308   OHOSPICU    : 387
                         (Other)     : 225
```

Gastrointestinal surgery

```
class   VISIT    gender  emerg   aids    myeloma  lymphoma tumorwm  immunosup
 0:8546  1:8469   0:4488  0:6003  0:8959   0:8917   0:8910   0:8122   0:7964
 1: 429   2: 506  1:4487  1:2972  1:  16   1:  58   1:  65   1: 853   1:1011

 hepfail  copd     diabetic DIALYSIS
 0:8675   0:7738   0:6655   0:8697
 1: 300   1:1237   1:2320   1: 278

              ethnic           IPRIORLOC
 African.American:1030   FLOOR.OTHER:   1
 Asian           : 110   ICUTRANS    :  67
 Caucasian       :7398   OPROOM      :3987
 Hispanic        : 115   RR          :4920
 Native.American :  96
 Other.Unknown   : 226
```

Trauma diagnosis

```
class   VISIT    gender  aids    myeloma  lymphoma tumorwm  immunosup hepfail
 0:6441  1:6836   0:4506  0:6861  0:6839   0:6856   0:6784   0:6767    0:6836
 1: 427  2:  32   1:2362  1:   7  1:  29   1:  12   1:  84   1: 101    1:  32

 copd     diabetic DIALYSIS
 0:6432   0:5895   0:6786
 1: 436   1: 973   1:  82

              ethnic           IPRIORLOC
 African.American: 663   ED          :5509
 Asian           : 133   OHOSP       : 566
 Caucasian       :5335   HOME        : 409
 Hispanic        : 234   FLOOR.OTHER: 178
 Native.American :  91   SDU         : 100
 Other.Unknown   : 412   OHOSPICU    :  42
                         (Other)     :  64
```

Trauma surgery

```
class   VISIT   gender  emerg   myeloma lymphoma cirrhosis tumorwm  immunosup
 0:2119  1:2190  0:1472  0: 718  0:2254  0:2254   0:2230    0:2241   0:2233
 1: 142  2:  71  1: 789  1:1543  1:   7  1:   7   1:  31    1:  20   1:  28
```

```
 hepfail  copd     diabetic DIALYSIS
 0:2249   0:2067   0:1940   0:2229
 1:  12   1: 194   1: 321   1:  32

              ethnic        IPRIORLOC
 African.American: 260   ICUTRANS:   9
 Asian           :  43   OPROOM  :1129
 Caucasian       :1722   RR      :1123
 Hispanic        :  75
 Native.American :  47
 Other.Unknown   : 114
```

Metabolic diagnosis

```
class    VISIT   gender  aids    myeloma  lymphoma tumorwm  immunosup hepfail
 0:6670  1:6539  0:3344  0:6815  0:6787   0:6807   0:6656   0:6500    0:6757
 1: 169  2: 300  1:3495  1:  24  1:  52   1:  32   1: 183   1: 339    1:  82

 copd     diabetic DIALYSIS
 0:6134   0:2293   0:6184
 1: 705   1:4546   1: 655

              ethnic           IPRIORLOC
 African.American:1852   ED          :4558
 Asian           :  93   FLOOR.OTHER: 930
 Caucasian       :4488   OHOSP       : 692
 Hispanic        : 205   SDU         : 271
 Native.American :  48   TELEMETRY   : 198
 Other.Unknown   : 153   OHOSPICU    :  78
                         (Other)     : 112
```

Genitourinary diagnosis

```
class   VISIT   gender  aids    myeloma lymphoma cirrhosis tumorwm  immunosup
 0:2199  1:2239  0:1257  0:2326  0:2302  0:2315   0:2217    0:2212   0:2112
 1: 137  2:  97  1:1079  1:  10  1:  34  1:  21   1: 119    1: 124   1: 224

 hepfail  copd     diabetic DIALYSIS
 0:2275   0:1959   0:1331   0:2122
 1:  61   1: 377   1:1005   1: 214

              ethnic           IPRIORLOC
 Other.Unknown   :  63   ED          :1075
 African.American: 479   OHOSP       : 489
 Asian           :  32   FLOOR.OTHER: 445
 Caucasian       :1703   SDU         : 119
 Hispanic        :  42   TELEMETRY   :  73
 Native.American :  17   OHOSPICU    :  72
                         (Other)     :  63
```

Chapter 4 : Summary and future research

4.1 Summary

In this thesis, a new genetic algorithm framework for variable selection is proposed. The main advantage of the methodology described is that it can handle an arbitrary number of potential predictors, making it well-suited for Data Science problems and especially applications where the modeler suspects the presence of significant interaction terms. Many variable selection algorithms can struggle in high-dimensional problems and the inclusion of interaction terms expands the potential predictor space in a combinatorial fashion, which amplifies this weakness. To make matters worse, the detection of interaction terms is a complex problem that often boils down to a modeler's subjective judgement and subject matter knowledge. This process is once again greatly hindered by high-dimensional problems.

Aside from the difficulties posed by high-dimensional problems for many variable selection procedures, in many cases the benefits of including interaction terms in the model are relatively marginal compared to the contribution of main effects terms. Thus, in the interests of model parsimony the study of interaction terms has often been relegated to a secondary priority unless there is a strong presupposed basis for their inclusion. However, the increasing availability of larger and more complex datasets for predictive modeling also increases the likelihood that significant interaction terms may be found in such datasets, allowing the opportunity for models to improve predictive performance by actively searching for and including interaction terms.

The use of a genetic algorithm for variable selection also provides great flexibility to the modeler. The GA is able to operate without any assumptions regarding the problem space structure and the selection process is independent of the model used and the measure used for evaluating fitness. Thus, the modeler is able to choose a fitness measure that best suits their purpose (i.e. AUC, classification accuracy, adjusted R^2 etc.) and the variable selection process will directly optimize over the chosen fitness measure. This is in contrast to other variable selection methods which evaluate variables using a predefined statistical metric of variable informativeness, e.g. AIC, BIC, MSE. While such measures have proven to perform well in terms

of selecting informative variables, there is no guarantee that a model with optimal AIC will also have, for example, an optimal AUC score if that is the performance metric that is the modeler's primary concern.

In chapter 1, the GA framework is used on several small toy problems to demonstrate that the algorithm is able to find the optimal solution. In chapter 2, the methodology is applied to two high-dimensional medical datasets using logistic regression models (with AUC as a fitness function) and compared to logistic regression models with stepwise selection and random forest models. The GA selected models compared favorably to both stepwise selection and random forest models and were able to find significant interaction terms despite having a very large number of potential predictors. In chapter 3, the GA selection framework was used to develop a prototype ICU scoring system that included interaction terms. While data was unavailable for a comparison with existing ICU scoring systems, the new system achieved good discrimination, with a weighted average AUC of 0.84 across all diagnostic categories and markedly better performance (AUC > 0.9) in certain categories.

These studies show that the proposed GA formulation is able to effectively perform variable selection in high-dimensional datasets, and can help modelers in exploratory data analysis to find significant interaction terms which may lead to further insights into the problem being examined.

4.2 Future research

4.2.1 Improving algorithm run-time

One of the biggest concerns with using a genetic algorithm approach is that it is computationally expensive compared to other methods of variable selection. This does not pose a problem when the algorithm run-time is not a critical concern, for example when the variable selection process only needs to occur infrequently during model specification. However, it does mean that the GA framework suffers in a dynamic, time-critical environment. There exist multiple avenues for improving the GA's run-time. One possibility is in optimizing the code to take full advantage of massively parallel computation tools such as utilizing GPUs.

Another is in fine-tuning the GA's meta-parameters for each individual application, or exploring other schemes for recombination, mutation or selection. For example, instead of selecting variables to be included in the chromosome according to a uniform distribution, a weighted distribution function can be used to increase the likelihood of selecting interaction terms when at least one of the corresponding main effects terms is already in the chromosome. Using multiple recombination points could also increase the GA's exploration rate.

4.2.2 Model generalizability

Another concern with GA methods is that they are susceptible to over-fitting the training data. In this case, cross-validation was used in the fitness function to reduce this tendency. Nevertheless, the models returned by the GA are not necessarily generalizable to other datasets and further study is required to validate these models. In particular, the ICU scoring system models should be tested on another dataset of ICU patients, preferably with the ability to compare the scores for each patient with scores from other ICU scoring systems such as APACHE, SAPS or MPM.

4.2.3 Model simplification

The models returned by the GA selection method can be further refined as the selection method does not overtly penalize a large model size. Further study needs to be done to determine an appropriate penalty function to prune the resulting models without constraining the GA's ability to effectively search the solution space for predictors which increase model fitness.