A COMPARISON OF THE PREDICTIVE ACCURACY OF LASSO REGRESSION AND NEURAL NETWORK MODELS, INTERNALLY AND EXTERNALLY VALIDATED USING SIMULATED DATA

by

WILLIAM ARTHUR LINDBLAD, III

(Under the Direction of Kevin Dobbin)

ABSTRACT

In recent years, a great deal of biomedical research has been focused on identifying biomarkers that can be used in settings of clinical research and practice to evaluate exposure, effect, or susceptibility of a patient to external stimuli. One such area of research has sought to discover and classify biomarkers that can be used to guide treatment selection in patients, especially those with different types of cancer. Despite various attempts, there remains a lack of consensus about how to best objectively select candidate genes that could inform medical decisions to maximize the treatment outcome for patients. The purpose of this study was to investigate two common statistical methods used for prediction, and to compare them under simulated circumstances to evaluate their internal and external validity in a real-world clinical setting. Overall, the lasso regression model displayed a greater robustness to various levels of simulated variation compared to the neural network.

INDEX WORDS: Lasso, Neural Network, Internal Validation, External Validation, Simulation, Prediction

A COMPARISON OF THE PREDICTIVE ACCURACY OF LASSO REGRESSION AND NEURAL NETWORK MODELS, INTERNALLY AND EXTERNALLY VALIDATED USING SIMULATED DATA

by

WILLIAM ARTHUR LINDBLAD, III

B.S., The Georgia Institute of Technology, 2008

M.P.H., Armstrong Atlantic State University, 2010

A Thesis Submitted to the Graduate Faculty of The University of Georgia in Partial Fulfillment of the Requirements for the Degree

MASTER OF SCIENCE

ATHENS, GEORGIA

2017

A COMPARISON OF THE PREDICTIVE ACCURACY OF LASSO REGRESSION AND NEURAL NETWORK MODELS, INTERNALLY AND EXTERNALLY VALIDATED USING SIMULATED DATA

by

WILLIAM ARTHUR LINDBLAD, III

Major Professor:	Kevin Dobbin
Committee:	Stephen Rathbun
	Xiao Song

Electronic Version Approved:

Suzanne Barbour
Dean of the Graduate School
The University of Georgia
December 2017

DEDICATION

This thesis is dedicated to my wife, Morgan, and to my daughter, Gracie. They are my inspiration to work hard and give everything my all. It is my strong desire to make them proud.

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my Creator and Savior, Jesus Christ, for the surpassing grace and boundless blessings that He has lavished upon my life. He has provided me with a loving family and numerous opportunities to expand my knowledge and experience through higher education. All glory is rightly due to Him.

I would also like to thank my family for their support and encouragement throughout the process of earning this graduate degree. They have sacrificed regularly in order for me to meet deadlines and prepare for exams throughout my career as a graduate student.

Lastly, but certainly not least, I would like to express deep gratitude to my professors, who have invested in me and my classmates so that we can enter the professional realm with sufficient preparation in the form of knowledge, practical skills, and experience. I am specifically thankful to the time and effort that Dr. Kevin Dobbin has contributed in helping me along the process of completing this thesis.

TABLE OF CONTENTS

Page

CHAPTER 1

INTRODUCTION/LITERATURE REVIEW

It has been known for a while that research into biomarker development has a strong impact on the effectiveness of newly designed drug therapies, making them more efficient and contributing to better outcomes (Taube et al., 2009). In addition to predicting the efficacy of treatment to improve clinical outcomes, biomarkers also have the capacity to decrease medical costs by specifically tailoring treatments to patients, which prevents wasted time and resources (Janes et al., 2015). Predictive biomarkers, also known as treatment selection markers, are factors that help clinical practitioners choose therapies that will maximize the positive health outcomes for their patients while minimizing risk of adverse effects (Janes et al., 2011). Despite the intense interest in discovering biomarkers related to cancer development and treatment, there are only a very limited number of clinically useful markers. Typically, early studies report that identified biomarkers are promising, but subsequent assessment yields inconclusive or contradictory results. This repeated discrepancy eventually led to the adoption of guidelines for the reporting of tumor marker studies to promote a standardized methodology for evaluating their usefulness (McShane et al., 2005). Simon also highlighted many obstacles that needed to be overcome to allow predictive biomarkers to achieve their potential in developing more effective treatments and tailoring them to specific subsets of patients that would most benefit from their use. One such limitation was the resistance to inter-disciplinary collaboration, while other hurdles included the need for innovation in the design and implementation of drug development and clinical studies (Simon, 2008).

Despite increased focus on developing biomarkers that predict treatment response and can therefore inform a patient's treatment decisions, there has not been a corresponding expansion of statistical methodology to objectively evaluate them for accuracy (Janes et al., 2014). A number of older statistical methods exist to gauge a predictive biomarker's utility: assessing prognostic value, examining treatment effects in groups of patients with restricted biomarker values, and testing for interactions between a selected treatment and the biomarker values of the patients. These methods, however, are not sufficient because of their limited scope and generalizability (Janes et al. 2011). One well known model that employs predictive biomarkers to guide treatment selection is the Gail breast cancer prediction model, which identifies older women who may benefit from tamoxifen-based prevention strategies rather than be harmed by them. This type of model is often used to identify high-risk subsets of patients, and has typically been paired with a statistical measure of treatment effect to evaluate the impact of treating those subsets. Alternatively, other approaches have used data obtained from randomized clinical trials to model treatment effect on an outcome measure that includes positive and negative impacts (Janes et al., 2013). Furthermore, standard measures for diagnostic tests, such as sensitivity, specificity, and predictive value (positive and negative) have been proposed to assess the usefulness of predictive biomarkers, with some modification, but they still require a number of assumptions to be applicable and interpretable (Simon, 2015).

Given the lack of consensus in the literature regarding standards of statistical methodology to evaluate the predictive accuracy of treatment selection biomarkers, this study will attempt to address that gap in current knowledge by suggesting and comparing two commonly used methods of prediction: lasso regression and artificial neural networks. These methods will be applied to simulated data and validated both internally and externally. Their

respective predictive accuracies will be compared under a variety of conditions to determine whether increased variation in theoretical samples consistently and differentially influences the performance of one model more than another. The results of these comparisons may then be used to provide insight and direction toward establishing a universally accepted methodology for evaluating prediction accuracy.

Lasso Regression

Lasso (or, as an acronym, LASSO), also referred to as regression with L1 regularization, stands for "least absolute shrinkage and selection operator," and in the discipline of statistics, it is a type of regression analysis technique developed by Robert Tibshirani in the mid-1990s that enhances the prediction accuracy of general regression models by facilitating variable selection and regularization (Tibshirani, 1996). This is accomplished by altering the model fitting process to select only a subset of covariates for use in the final model instead of including all of them. It was originally developed for use with least squares models, a simplified case, but was eventually extended to a larger array of models, such as generalized linear models, generalized estimating equations, and even Cox proportional hazard models.

Before the lasso was developed, the most common method for final variable selection was stepwise selection, but this was limited in its ability to improve prediction accuracy in cases where there were not a set of covariates with strong correlation to the outcome of interest. In fact, there are a number of scenarios where the stepwise process would lead to greater error in prediction. At the time, ridge regression was the ubiquitous procedure for increasing predictive accuracy, and it operated by shrinking regression coefficients to decrease the amount of overfitting present in the model. The lasso technique combined the strengths of these into a single process, which forces the sum of the absolute value of the regression coefficients to be

less than a predetermined value. This necessarily leads some coefficients to become zero in value, and excluding them from the resulting model. Mathematically, this means that the lasso process can be described by the following equation:

$$\min(\beta_0, \beta)\left\{\frac{1}{N}\sum_{i=1}^{N}(y_i - \beta_0 - x_i^T\beta)^2\right\} \text{ subject to } \sum_{j=1}^{p}\left|\beta_j\right| \leq t$$

In a sample of N observations, each of which have p covariates and a single outcome identified as y_i, x_i is defined as the covariate vector of the i^{th} observation. The objective of the lasso procedure is to solve the equation above, when t is a value representing the extent of regularization.

Since its original development, a number of extensions and variations have been derived for use in particular situations. A generalization known as the elastic net incorporates a secondary penalty similar to that found in ridge regression (Zou and Hastie, 2005). This addition has the effect of improving prediction performance when the number of predictors exceeds the sample size of a data set, as the lasso can only select a number of covariates equal to or less than the sample size. This would make the lasso a special case of the elastic net method. Likewise, a procedure called the adaptive lasso was developed which is not a special case of the elastic net (Zou, 2006), and it differs from the lasso in that it possesses a characteristic known as the oracle property. The oracle property describes a situation wherein the predictive model performs as well as if the true underlying model were given in advance.

Neural Networks

Artificial neural networks (often simply "neural networks") have been described as computing systems inspired by the biology of neural networks that comprise human and animal brains. These systems progressively improve their performance in prediction and classification

by “learning” from previous iterations of calculations and making adjustments. Neural networks are composed of a collection of units known as “neurons,” and each connection between them can transmit information. These connections, in turn, can have a weight assigned to them as the model converges to a solution, which can increase or decrease the extent to which a particular neuron affects the outcome.

In the 1990s, artificial neural networks first began being applied to questions of prediction and classification, which had traditionally been the domain of regression modeling (Warner & Misra, 1996). Specifically, neural networks represent a decision-making aid for clinicians, allowing for the processing of large quantities of interrelated data (Cross, Harrison, & Kennedy, 1995). In effect, neural networks behave similar to a nonparametric regression model, which allows for a more robust functionality (Warner & Misra, 1996).

In terms of structure, neural network models are collections of interconnected elements termed “neurons.” These neurons receive input, change their state (a process known as “activation”), and produce a resulting output. The neurons are interconnected in a “network” arrangement such that some neurons’ outputs become inputs for others, forming a directed, weighted graph. The weighting of the individual connections can change based on prespecified conditions, allowing the network to improve its predictive performance in a process known as “learning.” Typically, the neurons are classified into distinct conceptual layers. The input layer is typically composed on the predictors of a data set, which serve as the starting point for the network computation. Then, there is a hidden layer composed of neurons which receive the outputs from the input layer neurons, each with its own weight. There may be more than one hidden layer in a neural network depending on the complexity of the data being analyzed, with the second and subsequent hidden layers taking the output of the previous layer as inputs. The

final hidden layer consolidates its neurons' outputs toward an output layer, which is typically a single neuron (corresponding to a single response variable), though there may be more in certain cases. It is a common rule of thumb, though not a necessity that the number of hidden layer neurons be between the number of input layer neurons and the number of output layer neurons. **Figure 1** below is a representation of typical neural network with ten input neurons, a single hidden layer composed of one neuron, and a single output neuron.

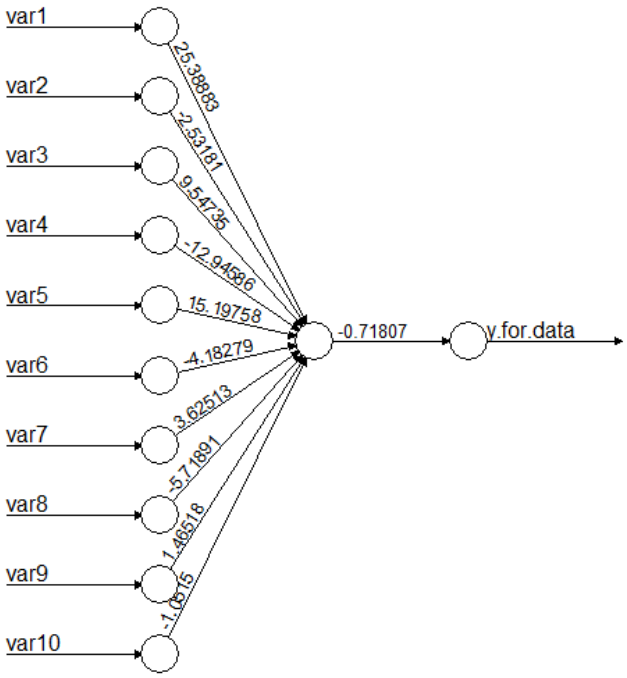

Figure 1 – A Typical Neural Network as Used in This Study

CHAPTER 2

METHODS

There were two main procedures that were used to maximize the extent to which the simulations were informative: internal and external validation. Both are described in greater detail in the subsequent sections, but in summary, internal validation was conducted by simulating variation from person to person, while external validation was conducted by simulating variation between different labs that may be involved in collecting and preparing biomarker samples. All simulations were run using R, a programming language used primarily for statistical computing. The *glmnet* package (*ver. 2.0-13*) was used to compute the lasso regression model, and the *neuralnet* package (*ver. 1.33*) was used to produce the neural network model.

Internal Validation

The first step in this procedure was to use the *MASS* package to generate the random data to be used in the simulation. To simplify computation (and therefore produce adequate results in a time-efficient manner), 100 sample data points were generated from a multivariate normal distribution, 50 from a group deemed "Class 1" and 50 from a group deemed "Class 2." Class 1 data points were created using a mean vector of ten dimensions, where the first element was set equal to 1, and all other elements held equal to 0. Class 2 data points were generated in similar fashion, but with the first element set equal to -1. The covariance matrix used for both classes of data was the identity matrix. These 50 observations from each class were then merged into a single data set of 100 total observations. Additionally, a set of 100 binary responses were also

generated for the data, with values being either 0 or 1, and this was to serve as a classifier for the observations, simulating a yes/no or positive/negative outcome.

Following the generation of the simulation data, a k-fold cross-validation procedure was implemented via the *glmnet* package, which returned an array of lambda values. Using the minimum lambda value, a generalized linear model was fit via a penalized maximum likelihood model using the lasso penalty. At this point, a validation set was generated with 1000 observations, 500 representing Class 1 data points and 500 data points of Class 2. The lasso regression model was then used to predict the responses of the validation set. The results, recorded in a 2x2 table, were then used to calculate the predictive accuracy of the model (i.e., the number of responses the model correctly predicted divided by the total sample size of 1000). This entire process was then repeated with different values in the mean vector to gauge how the predictive accuracy changed with simulated data with increased and decreased differences in means between the two classes.

The same generated data from the first step of the procedure used to validate the lasso was also used to validate a neural network model. This data was converted into a data frame format, and the predictors and response were assigned labels for simplicity. These labels were then used to establish a regression equation to be used in training the neural network, and the network model was fit using a single hidden layer of one neuron. This was done for computational efficiency and to make it maximally comparable to the lasso regression model, as more hidden neurons would have increased computation time with little to no payoff in increased predictive accuracy for the simplified simulated data. Likewise, having a single hidden layer neuron causes the mathematical characteristics of a neural network to approximate a linear regression model, as all of the predictors are used to predict a single response. The prespecified

algorithm used to calculate the neural network employed resilient backpropagation with weight backtracking.

Once the neural network was fit, a plot was produced to visualize the connectivity of the nodes and to visually inspect the weights of the respective outputs. As with the lasso, an independent validation set of 1000 observations was then generated, and the neural network was evaluated for predictive accuracy based on correct and incorrect predictions in a 2x2 table of results. This process was repeated as before with differing values of positive and negative means to determine how the predictive accuracy of the neural network changed as the difference in the means increased in magnitude.

The resulting predictive accuracies of both the lasso regression and the neural network procedures were compiled and tabulated, and those percentages were plotted as accuracy curves. Comparisons between the two models were then made numerically and by visual inspection of the plotted graphic. This procedure is summarized in the algorithm below:

Simulation Algorithm for Internal Validation in R

1. Load *MASS* package in R.
2. Generate 50 sample data points from a multivariate normal (MVN) distribution with mean vector and identity covariance. The mean vector contains 10 elements, the first being prespecified for the simulation and the other 9 being set equal to 0.
3. Generate another 50 sample data points from a MVN distribution with mean vector and identity covariance. The mean vector contains 10 elements, the first being prespecified for the simulation as a negative of the previous step and the other 9 being set equal to 0 as before.
4. Merge the simulated data points into a 100-observation array.
5. Create a binary response variable to be a classifier.
6. Load the *glmnet* package and perform a cross-validation procedure to produce a vector of lambda values. Choose the lowest value to use.
7. Fit a lasso regression model using the chosen lambda value.
8. Create a validation set by repeating steps 2-4 above, and increasing the number of data points in each group from 50 to 500. There will be 1000 total observations in the validation set.

9. Use the *predict()* function in the *glmnet* package to predict the response variable for each observation.
10. Tabulate the results of the prediction procedure, and determine accuracy by dividing correctly classified results by 1000.
11. Repeat steps 1-10 with a different prespecified mean vector in the data generation step.
12. Using the data generated in steps 2-4, merge the data set with the simulated binary response values into a data frame.
13. Load the *neuralnet* package and train a network using the newly created data frame.
14. Create a regression equation to be used to by the neural network.
15. Fit and plot the neural network.
16. Use the *compute()* function to predict the response variable for each observation.
17. Tabulate the results of the prediction as in step 9.
18. Repeat steps 12-17 with each new set of data points with prespecified mean vector values.

External Validation

Additional steps were taken to externally validate the models under analysis using generated data points. The aforementioned internal validation procedures were implemented to simulate real-world context of predictive biomarkers, albeit in a highly simplified form. External validation procedures take that a step farther by simulating variation that would be anticipated by having different study sites or laboratories process data independently of one another. Thus, the internal validation procedure can be regarded as equivalent to a special case of external validation where the amount of variation across sites is set equal to 0. In order to model this for the lasso regression and neural network procedures, a certain amount of noise was artificially added to the validation sets generated by the computer software. Thus, instead of just merging two randomly generated 500-observation datasets and applying each model as before, now there would be 10 different sets to merge, composed of five "laboratories" each with positive and negative mean values. Moreover, each of the "laboratories" would have a randomly generated variability based on a multivariate normal distribution and a prespecified between-laboratory variance (called "tau-squared"). All of this would add a layer of complexity to predicting responses, and as a result of the increased variability, the process was looped to produce 10

iterations of each model's predictions, all of which were averaged to yield a mean accuracy for the model. This validation setup can be represented mathematically by the following pair of formulas:

For class 1, laboratory i, person j,

$x_{ij} = \mu + \tau * z_i + z_j$, where zi and zj are both multivariate normal with mean 0 and identity covariance, all independent.

For class 2, laboratory i, person j,

$x_{ij} = -\mu + \tau * z_i + z_j$, where zi and zj are both multivariate normal with mean 0 and identity covariance, all independent.

These formulas indicate that external validation took into account variability arising from both the laboratory setting and the intrinsic variation from person to person, while internal validation only incorporated the latter, as tau-squared was set to 0 in that instance.

As was the case with the internal validation procedures, the external validation process was repeated numerous times using various combinations of means and tau-squared, to produce comprehensive results for numerous scenarios. Each combination of prespecified parameters was tabulated and plotted for each model, and comparisons were made on the basis on accuracy percentages and visual inspection of the accuracy curves. As with internal validation, external validation can be summarized in the following algorithm:

Simulation Algorithm for External Validation in R

1. Load *MASS* package in R.
2. Generate 50 sample data points from a multivariate normal (MVN) distribution with mean vector and identity covariance. The mean vector contains 10 elements, the first being prespecified for the simulation and the other 9 being set equal to 0.
3. Generate another 50 sample data points from a MVN distribution with mean vector and identity covariance. The mean vector contains 10 elements, the first being prespecified

for the simulation as a negative of the previous step and the other 9 being set equal to 0 as before.

4. Merge the simulated data points into a 100-observation array.
5. Create a binary response variable to be a classifier.
6. Load the *glmnet* package and perform a cross-validation procedure to produce a vector of lambda values. Choose the lowest value to use.
7. Fit a lasso regression model using the chosen lambda value.
8. Create a validation set by repeating steps 2-4 above, but instead of just generating 2 groups of data points, 10 groups will be created to represent 5 study sites, each with it's own 2 groups. Also, this is where inter-site variability is introduced via specifying a value of tau-squared to incorporate into the randomly generated MVN data.
9. Use the *predict()* function in the *glmnet* package to predict the response variable for each observation.
10. Tabulate the results of the prediction procedure, and determine accuracy by dividing correctly classified results by 1000.
11. Repeat steps 1-10 10 times and take the mean accuracy value.
12. Repeat steps 1-11 with a different prespecified mean vector in the data generation step.
13. Using the data generated in steps 2-4, merge the data set with the simulated binary response values into a data frame.
14. Load the *neuralnet* package and train a network using the newly created data frame.
15. Create a regression equation to be used to by the neural network.
16. Fit and plot the neural network.
17. Use the *compute()* function to predict the response variable for each observation.
18. Tabulate the results of the prediction as in step 9.
19. Repeat steps 14-18 10 times and take the mean accuracy value.
20. Repeat steps 14-19with each new set of data points with prespecified mean vector values.

CHAPTER 3

RESULTS/DISCUSSION

Internal Validation

The results of the internal validation procedure are shown in **Table 1** and **Figure 2**.

Table 1 - Comparison of Model Accuracy: Internal Validation Procedure			
Mean (+/-)	**Difference**	**Lasso Accuracy**	**Neural Network Accuracy**
0.25	0.50	56.4%	57.4%
0.50	1.00	62.2%	60.8%
0.75	1.50	77.4%	66.9%
1.00	2.00	84.6%	76.2%
2.00	4.00	96.7%	93.9%
3.00	6.00	99.9%	99.2%
4.00	8.00	100.0%	100.0%

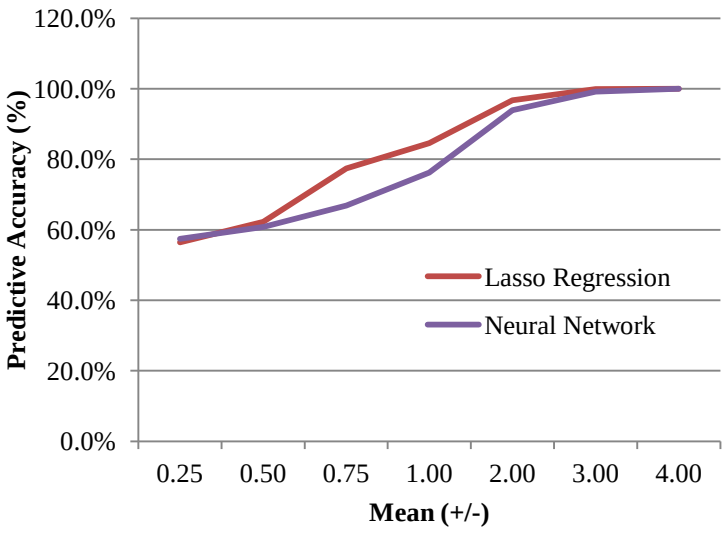

Figure 2 – Plot of Accuracy Curves for Lasso Regression and Neural Network Under Various Differences in Mean

Both the lasso regression model and the neural network appear to increase in predictive accuracy as the differences in mean become larger. This is to be expected, as the more spread apart the means of the two groups become, the more easily distinguishable they become from one another, allowing for a more accurate classification. By contrast, groups where the respective means are closer together are less easily classified, so each model's accuracy is lower in that scenario. Based on the percentages in **Table 1**, and the corresponding plot in **Figure 2**, the lasso model and neural network have similar accuracy at the extreme ends of the tested mean values. When the difference in group means is only 0.5, the lasso regression produces an accurate classification 56.4% of the time compared to the 57.4% accuracy of the neural network. Likewise, when the difference in mean values is increased to 8.0, both models achieve a 100% accuracy. The main difference to be found in the performance of the respective models is the fact that the lasso has a slightly more rapid rise in accuracy across the range of simulated values, which gives it the edge over the neural network. This result is also to be expected based on the conceptual basis for each model. Despite the inclusion of ten predictor variables, only the first one is informative with respect to the response variable. The lasso procedure is more efficient at removing predictors that are not highly related to the outcome of interest, forcing their regression coefficients to take on a value of 0. In comparison, the neural network maintains all ten predictor variables, even while nine of them are non-informative regarding the outcome of interest. By leaving them in the model, the accuracy is reduced while the network recalculates the weight each input variable should be given to optimize the output.

External Validation

The results of the external validation procedures can be found in **Tables 2-6** and **Figures 3-7**, each of which corresponds to a particular value of tau-squared prespecified in the

simulation. For the tables and figures, the respective values of tau-squared are as follows: 0.01, 0.10, 0.50, 1.0, and 2.0. Those values correspond to increasing variability in the validation set values, which in turn reflects the potential for variation between laboratories processing similar data.

Table 2 - Comparison of Model Accuracy: External Validation Procedure, Tau-Squared = 0.01			
Mean (+/-)	**Difference**	**Lasso Accuracy**	**Neural Network Accuracy**
0.25	0.50	57.7%	52.4%
0.50	1.00	60.1%	65.3%
0.75	1.50	75.7%	67.2%
1.00	2.00	83.1%	75.6%
2.00	4.00	97.0%	96.2%
3.00	6.00	99.7%	100.0%
4.00	8.00	100.0%	100.0%

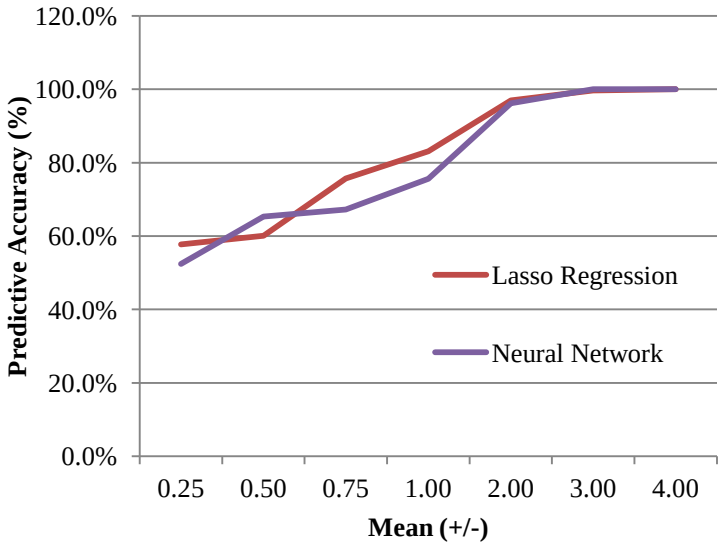

Figure 3 – Plot of Accuracy Curves for Lasso Regression and Neural Network Under Various Differences in Mean, Tau-Squared = 0.01

A value of tau-squared equal to 0.01 represents comparatively low variability between separate study sites processing the same information, as shown in **Table 2** and **Figure 3**. In the cases where the differences in means are relatively small, there is some equivocation between the models' respective accuracies, with neither being consistently superior. While both models display an expected trend of increasing predictive accuracy corresponding to increasing distance between group means, there is no clear superior in this instance as there was in the internal validation step. The lasso does manage to outperform the neural network at differences in the group means of 0.75 to 1.0, but the neural network quickly matches and briefly outperforms the lasso as they both approach 100%.

Table 3 - Comparison of Model Accuracy: External Validation Procedure, Tau-Squared = 0.10

Mean (+/-)	Difference	Lasso Accuracy	Neural Network Accuracy
0.25	0.50	53.2%	53.3%
0.50	1.00	67.0%	59.1%
0.75	1.50	76.4%	68.8%
1.00	2.00	83.4%	80.6%
2.00	4.00	95.8%	95.9%
3.00	6.00	99.6%	99.3%
4.00	8.00	100.0%	100.0%

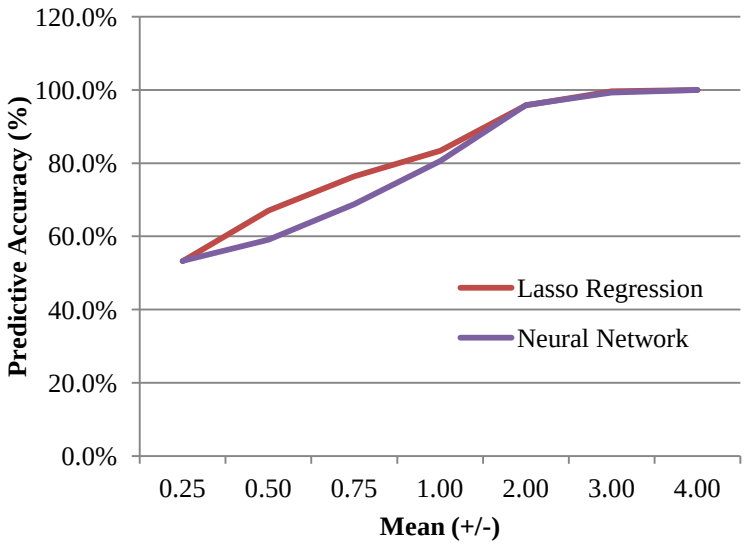

Figure 4 – Plot of Accuracy Curves for Lasso Regression and Neural Network Under Various Differences in Mean, Tau-Squared = 0.10

Table 3 and **Figure 4** correspond to the external validation procedure where tau-squared was set equal to 0.10, which is still a relatively low variability, but nevertheless it is ten times the previous value. Generally, this situation mirrors that of the internal validation results in that both models seem to have comparable predictive accuracy at both ends of the tested values, but with lasso having a slightly more rapid increase in accuracy compared to the neural network. However, in this instance, with noise and variability added to the validation set, the neural network and lasso seem to converge in accuracy a bit sooner than in previous attempts.

Table 4 - Comparison of Model Accuracy: External Validation Procedure, Tau-Squared = 0.50			
Mean (+/-)	**Difference**	**Lasso Accuracy**	**Neural Network Accuracy**
0.25	0.50	51.6%	53.2%
0.50	1.00	65.2%	51.3%
0.75	1.50	66.4%	65.2%
1.00	2.00	78.7%	77.4%
2.00	4.00	94.2%	92.8%
3.00	6.00	99.6%	99.1%
4.00	8.00	100.0%	99.9%

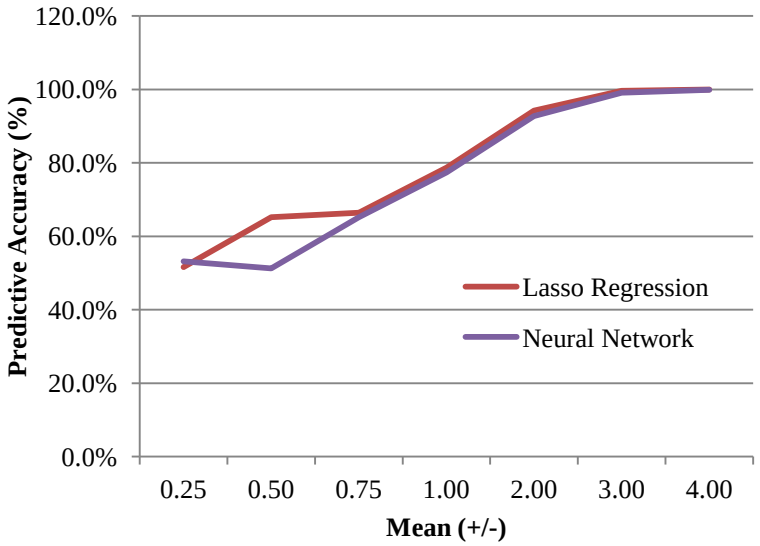

Figure 5 – Plot of Accuracy Curves for Lasso Regression and Neural Network Under Various Differences in Mean, Tau-Squared = 0.50

At a tau-squared level of 0.5, the predictive accuracies of the two models seem to fall in line with the results when tau-squared was set to 0.1. However, in this case, the five-fold increase in variability seems to correspond to a further lowering of the mean difference value at which the models achieve similar levels of accuracy. After starting off relatively equivalent in predictive

power, the lasso experiences a moderate increase in accuracy, while the neural network displays a modest decrease. This is immediately followed by a situation where the neural net then increases its accuracy while the lasso regression model remains about the same. The net result is that the models each approach the other in terms of accuracy, and remain almost identical until they both achieve approximately 100%.

Table 5 - Comparison of Model Accuracy: External Validation Procedure, Tau-Squared = 1.0			
Mean (+/-)	**Difference**	**Lasso Accuracy**	**Neural Network Accuracy**
0.25	0.50	49.6%	59.6%
0.50	1.00	63.4%	66.0%
0.75	1.50	72.5%	72.5%
1.00	2.00	75.8%	77.5%
2.00	4.00	90.1%	85.7%
3.00	6.00	99.7%	99.5%
4.00	8.00	99.8%	99.6%

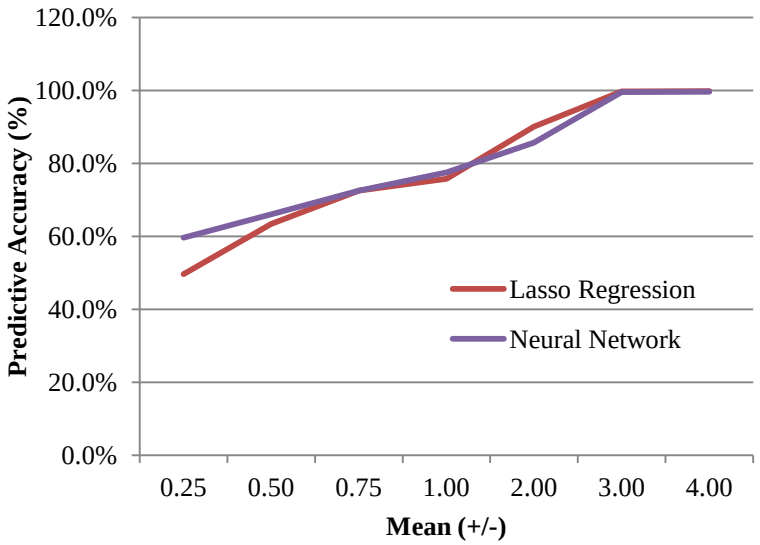

Figure 6 – Plot of Accuracy Curves for Lasso Regression and Neural Network Under Various Differences in Mean, Tau-Squared = 1.0

The trend of the lasso regression and neural network having very similar accuracy curves is continued in **Table 5** and **Figure 6**, which display the results of external validation when tau-squared is set to 1.0. However, there is some variation to note, particularly at the lower end of the observed accuracies. The neural network seems to outperform the lasso in the instance of very close group means (59.6% vs. 49.6%). As the group means get farther and farther apart, making them more distinguishable, the two models seem to have similar ability to correctly predict the response based on the predictors. Based on the percentages and the corresponding plot of the accuracies, the two models seem to have a very similar performance at this level of variation. Also, interestingly, this is the first time where neither the lasso regression nor the neural network were not able to achieve a full 100% accuracy.

Table 6 - Comparison of Model Accuracy: External Validation Procedure, Tau-Squared = 2.0

Mean (+/-)	Difference	Lasso Accuracy	Neural Network Accuracy
0.25	0.50	46.3%	51.5%
0.50	1.00	69.9%	73.3%
0.75	1.50	61.9%	63.7%
1.00	2.00	69.5%	64.9%
2.00	4.00	81.5%	86.7%
3.00	6.00	95.4%	82.8%
4.00	8.00	99.7%	99.9%

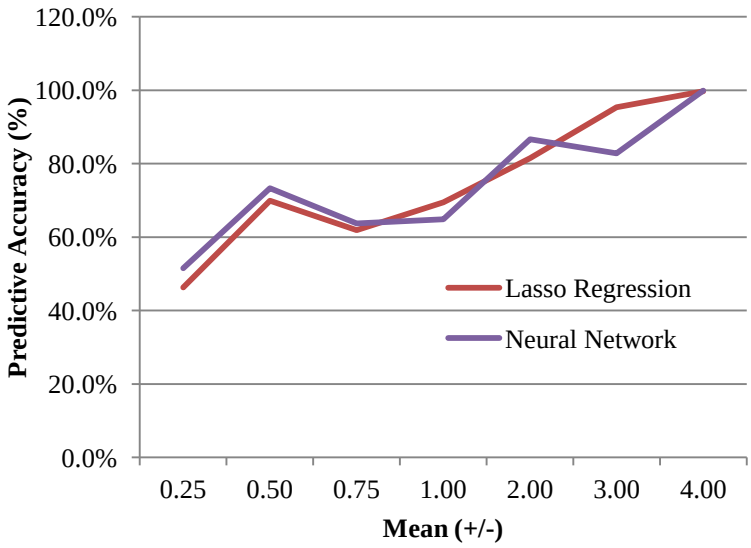

Figure 7 – Plot of Accuracy Curves for Lasso Regression and Neural Network Under Various Differences in Mean, Tau-Squared = 2.0

Table 6 and **Figure 7** represent the maximum amount of variability which was simulated in this study (tau-squared = 2.0), which is 200 times the lowest level of variability simulated in the external validation procedure. Fittingly, this amount of random variation has produced the noisiest plot of the accuracy curves in that the curves do not simply increase monotonically as in previous iterations, but they increase and decrease at least once each (twice in the case of the neural network). Likewise, this round of simulation continues the trend that began when tau-squared was set to 1.0 in that neither of the models was able to achieve 100% accuracy even with such a large difference in group means as 8.0.

Overall, both models performed well at the task of prediction. Early on, the lasso held the advantage over neural network in yielding more accurate results, especially in the internal validation step. However, in the external validation steps, when variation was introduced into the

simulation, the neural network seemed to gain ground on the lasso in predictive accuracy, and debatably overtook it at the higher levels of variability. That would seem to indicate that the neural network is a more robust system overall, especially in real-world scenarios when biomarkers are multifactorial in nature, and some amount of variability is a given. The higher the dimension of the data, the more a neural network would seem to have an advantage over lasso regression.

CHAPTER 4

CONCLUSIONS

Limitations

There are a number of limitations to this study, not the least of which is the use of pre-programmed simulated conditions as a proxy for actual clinical data. This study remains highly theoretical in its conclusions and observations due to the nature of the data that was generated for analysis. Likewise, only two models were being compared, and there remains myriad other statistical models that could be deployed toward the same purpose which were not included in this evaluation. Even within the lasso regression method, there are variations that could be of use in the case of prediction.

Future Directions

Most immediately, these simulated experiments should be carried out with many more replications for each combination of parameters, in a Monte Carlo framework. This would give more robust values for prediction accuracy at each level of intrinsic and extrinsic variation programmed into the simulations. Once those simulations are concluded, however, the ultimate test of the predictive accuracy of either model will be to use them on real-world data in a clinical trial setting. This will translate theoretical mathematical results to practical application.

REFERENCES

Cross, S.E., Harrison, R.F., & Kennedy, R.L. (1995). Introduction to neural networks. *Lancet*, 346, 1075-1079.

Janes, H., Pepe, M.S., Bossyut, P.M. & Barlow, W.E. (2011). Measuring the Performance of Markers for Guiding Treatment Decisions. *Ann Intern Med*, 154, 253-259.

Janes, H., Pepe, M.S., & Huang, Y. (2013). A Framework for Evaluating Markers Used to Select Patient Treatment. *Med Decis Making*, 34, 159-167.

Janes, H., Brown, M.D., Huang, Y., & Pepe M.S. (2014). An Approach to Evaluating and Comparing Biomarkers for Patient Treatment Selection. *Int J Biostat*, 10(1), 99-121.

Janes, H., Brown, M.D., & Pepe, M.S. (2015). Designing a study to evaluate the benefit of a biomarker for selecting patient treatment. *Stat Med*, 34(27), 3503-3515.

McShane, L.M., Altman, D.G., Sauerbrei, W., Taube, S.E., Gion, M., & Clark, G.M. (2005). Reporting Recommendations for Tumor Marker Prognostic Studies (REMARK). *J Natl Cancer Inst*, 97, 1180-1184.

Simon, R. (2008). Lost in translation: Problems and pitfalls in translating laboratory observations to clinical utility. *Eur J Cancer*, 44, 2707-2713.

Simon, R. (2015). Sensitivity, Specificity, PPV, and NPV for Predictive Biomarkers. *J Natl Cancer Inst*, 107, djv157.

Taube, S.E., Clark, G.M., Dancey, J.E., McShane, L.M., Sigman, C.C., & Gutman, S.I. (2009). A Perspective on Challenges and Issues in Biomarker Development and Drug and Biomarker Codevelopment. *J Natl Cancer Inst*, 101, 1453-1463.

Tibshirani, R. (1996). Regression Shrinkage and Selection via the Lasso. *J R Statist Soc B*, 58(1), 267-288.

Warner, B. & Misra, M. (1996). Understanding Neural Networks as Statistical Tools. *Am Stat*, 50(4), 284-293.

Zou, H. & Hastie, T. (2005). Regularization and Variable Selection via the Elastic Net. *J R Stat Soc*, 67(2), 301-320.

Zou, H. (2006). The Adaptive Lasso and Its Oracle Properties. *J Am Stat Assoc*, 101(476), 1418-1429.