3-17-2004

Prediction of Commuter Choice Behavior Using Neural Networks

Aaron L. Gregory
University of South Florida

Prediction of Commuter Choice Behavior Using Neural Networks

by

Aaron L. Gregory

A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science in Computer Science
Department of Computer Science and Engineering
College of Engineering
University of South Florida

Major Professor: Rafael A. Perez, Ph.D.
Lawrence O. Hall, Ph.D.
Sudeep Sarkar, Ph.D.

Date of Approval:
March 17, 2004

Keywords: transportation, radial basis, Gaussian, estimation

Table of Contents

List of Tables

List of Figures

Prediction of Commuter Choice Behavior Using Neural Networks

Aaron L. Gregory

ABSTRACT

In order to reduce air pollution and reduce the amount of traffic on highways in the western United States, certain states have set up worksite trip reduction programs. Employers in these states must comply with worksite trip reduction laws and submit trip reduction plans to their respective regulatory agency each year. These plans are currently evaluated manually, and are either rejected or accepted by the agency. There are two major flaws in this system; the first is the amount of time required by the agency to review a plan could be a matter of months, and the second is that human reviewers have subjective opinions regarding the effectiveness of plans.

The purpose of this thesis is to develop computer models using Radial Basis Function neural networks, with centers built using the k-means clustering algorithm. These networks will be compared against the performance of a commercial neural network-modeling program known as Predict, as well as the traditional method of selecting RBF neurons from the training set.

Chapter 1

Introduction

In order to reduce air pollution, and the amount of traffic on highways in the western United States (especially in California), certain states have set up Worksite Trip Reduction programs. These programs require that businesses with a certain number of employees at a single worksite implement programs to encourage employees to use alternative modes of transportation. Alternative modes of transportation consist of any mode of transportation other than driving alone. With most of these programs the responsibility for employees commuting behavior rests with their employer. Some jurisdictions have the authority to fine employers if their trip reduction programs do not meet the goals set by the government.

Employers are encouraged to provide incentives to promote the use of alternative transportation modes by their employees. Employers submit their worksite trip reduction plans to the respective government agency every year. These plan submissions detail the current mode split of the worksite, and the incentives the employer intends to offer for the upcoming year. Plans are reviewed by the agency, and either approved or rejected. Approval is based on whether the reviewer believes that the submitted plan will help the worksite reach its trip reduction goal.

There are two major problems with the current system. The first problem is that during peak submission times, the turnaround time can be on the order of a couple of months. This kind of turnaround time is unacceptable; because an employer is already a couple of months into the plan when they find out their plan was rejected. The second problem is that plan approval is based on the opinion of the reviewer, and a plan that

appears perfectly acceptable to one reviewer, might seem unacceptable to another reviewer.

A computerized model should be able to solve both of these problems. Since computer models are objective, the model will be able to provide a non-biased opinion on the potential of a certain program. Also the turnaround time could be reduced from a couple of months to perhaps a couple of days, depending on how fast the data from the plan is entered into the database. The government agencies could release this model to the employers in the area, so that their plans could be self-evaluated before submission. This self-evaluation could help employers quantify how effective a particular incentive is, thereby justifying the cost of the incentives. Employers could save money by dropping costly and ineffective plans and adding some more effective and cheaper plans if they exist.

There currently is no expertise in how effective specific incentives are; therefore the model will have to be built from available data. There currently models that were built using linear regression techniques, however the performances of these models are not very impressive.

The purpose of this thesis is to study the effectiveness of neural networks in predicting the effect that incentives have on worksite trip reduction programs. This thesis will evaluate Radial Basis Function neural networks against classical multi-layer perceptron networks built using Predict. In addition this thesis will evaluate the use of non-uniform sigma size in Radial Basis Function neural networks and their effectiveness on worksite trip reduction programs.

Chapter 2

Analysis and Transformation of the Data

2.1 Vehicle Trip Rate

Vehicle Trip Rate (VTR) is one commonly used metric for Worksite Trip Reduction programs, and is defined as the number of vehicles used transporting 100 employees to and from work. This metric is considered to be the most practical among transportation experts, because a 1-point reduction in VTR, equates to 1 vehicle being eliminated per 100 employees.

Another common metric used in the evaluation of worksite trip reduction is Average Vehicle Ridership (AVR). AVR is defined as the average number of employees in each vehicle used to commute to and from the worksite.

For consistency the VTR metric will be used. AVR values can be converted to VTR values by simply dividing 100 by the AVR value. For example an AVR of 1.0 is equal to a VTR value of 100 and an AVR of 2.0 is equivalent to a VTR value of 50.

2.2 Description of the Data

The data to be modeled was collected by local Air Quality Boards; these organizations receive the data from employers in paper form. An employer's worksite trip reduction program is required to submit a plan to the agency for every year. This paper form is what a plan reviewer analyzes when approving or denying a plan proposal. These plans provide information about the types of incentives an employer will offer for the following year; also the plan includes a financial estimate on how much the employer

expects to spend on their trip reduction program that year. In most jurisdictions these plan submissions are entered into a database for the purpose of data analysis by the agencies.

All of the databases received from the regulatory agencies came in Microsoft Access format. The first task was to convert the Access databases into tab delimited text files that the network building software can read, and construct the training and testing sets.

2.3 Los Angeles Data

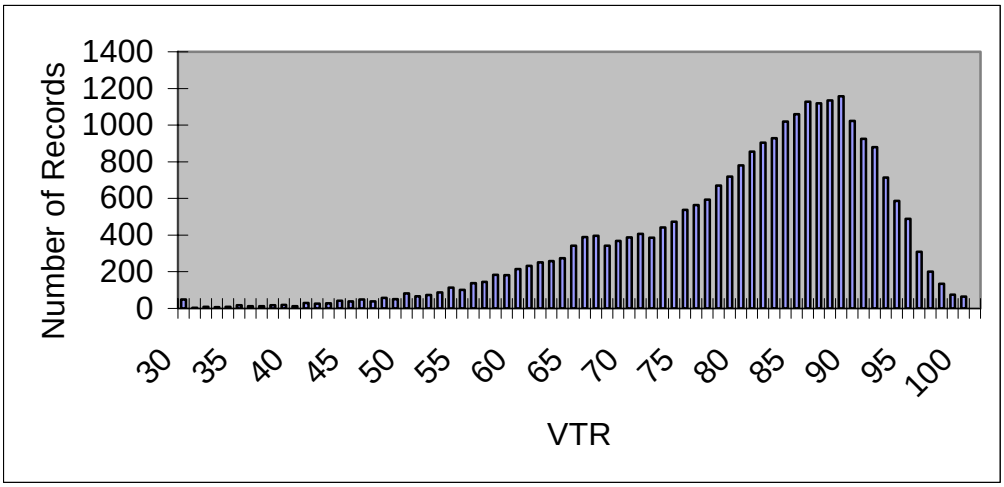
The Los Angeles Data has been collected by South Coast Air Quality Management District (SCAQMD) pursuant to a California law known as Rule 2202. This law requires that employers with more than 50 employees at a single worksite implement a trip reduction plan to reduce the number of vehicle trips to that worksite. Worksites with a VTR of 67 or less are exempted from submitting a plan.

The data from Los Angeles consists of 33,094 plans from 1990-2001. Each plan consists of the current year's mode split, incentives, and VTR. The incentives that are recognized by the regulatory agency are listed in Table 2.2. The mode split information consists of the number of persons who commute using the following modes of transportation; single occupant vehicle, 2-person carpool, 3-person carpool, 4-person carpool, 5-person carpool, 6-person carpool, vanpool, transit, bicycle, and walk. Mode split information includes the number of employees at the worksite on the day of the survey [10].

Not all of these 33,094 records in the database were useful, because the change in VTR was unknown for the final year's plan submission of each work site. Therefore the last year of submission for a worksite had to be deleted. After cleaning the data and calculating VTR the data set consisted of 25,043 records.

One particular problem immediately noticed with the data is that there seemed to be a few outliers. One example of an outlier is a particular worksite had a change in VTR of –20, which is considered to be a monumental improvement. Since the data is entered by hand from a form, this could be a sign that the data was slightly noisy. This should not be a problem since neural networks are able to adapt to moderately noisy data [2].

Figure 2. 1 VTR Histogram for Los Angeles Data



2.3.1 Data Conversion

The first step in converting this database into a usable form was to merge the different tables into a single "Master Table". All of the tables had a common primary key in common, so that each row in one table could be associated with row(s) from another table, therefore merging these tables was very straightforward. The primary key used to merge these tables was the combination of the "PermID", and "PlanYear" attribute. The "PermID" attribute is a unique identifier for a worksite, which is constant between the different years. The "Plan Year" attribute is an integer number, which is an offset from the baseline year (the baseline year is the first year the employer was required to report its trip reduction strategies).

2.4 Acceptable Ranges

For a result to be considered accurate, the neural network model should predict how the plan is going to perform for the upcoming year. The exact change is not as important as knowing whether or not a plan will exhibit a positive change and to what degree (small change, medium change, or high change). This is important, because if a worksite were far short of their goal, a larger change would be required in order for them to have an acceptable plan.

With the idea of acceptable ranges, there are two different views of this problem on how it relates to model building. The first is to view it as a prediction problem and try to calculate the exact change, and then convert it to whether the change is in the required range. Another is to view it as a classification problem, and try to predict which range each record will fall into. Here it is treated as a classification problem with the following classes; $-\infty$ to –20 (bin 1), -20 to –10 (bin 2), -10 to –3 (bin 3), -3 to –2 (bin 4), -2 to –1 (bin 5), -1 to 0 (bin 6), 0 to 1 (bin 7), 1 to 2 (bin 8), 2 to ∞ (bin 9). These bin numbers will be used when measuring performance in Chapter 4.

2.5 Attribute Selection

One key to building an accurate model is to select the proper attributes, because attributes that are extremely noisy or irrelevant can degrade the performance of the model. One group of attributes that must be selected is the incentive plans. Another attribute that should is the previous year's VTR, because if an employer has a previous VTR of 70, then their current VTR should be less than a company that started off with a previous VTR of 90. Also the change in VTR for the first company would usually be smaller, because it is much more difficult to make a VTR change from 70 to 65, than it is to make a VTR change from 90 to 85.

To determine which attributes would be selected for network building, the "variable selection" feature of a software package called Predict was used. Predict is an

application that builds neural networks and will be explained further in a later chapter. This "variable selection" feature uses statistical methods such as correlation to determine which attributes have an effect on the target value; it also helps in removing noisy and incomplete data.

2.6 Grouping Incentive Plans

Grouping similar incentive plans reduced the number of incentive plans from 64 plans to 14. One major reduction came in the form of reducing the number of guaranteed ride home programs, which is a plan where if an employee who used an alternative mode of transportation needs a ride home, that the employee would not be stuck at work, because their car is at home. This type of plan has many different types of implementation; the different guaranteed ride home programs recognized by the SCAQMD are shown in Table 2.1.

Table 2. 1 Example of Incentive Plans that can be Combined

Taxi Ride Home	Employer will provide a cab ride home.
TMA/TMO Ride Home	Transit Management Agency/Organization will provide a ride home.
Company Vehicle Ride Home	The company provides a ride home.
Unscheduled Overtime Ride Home	A ride home is provided in the case of unscheduled overtime.
Emergency Ride Home	A ride home is provided only in emergencies.
Other	Other type of ride home program

All of these different types of ride home programs are very similar, and combining these incentives into a single incentive could greatly reduce the time it takes to train the network. Another type of incentive that seemed to be replicated were different kinds of marketing strategies. SCAQMD recognized 14 different types of non-financial marketing strategies; an attribute was also created which grouped together all non-financial marketing incentives.

Grouping these similar types of incentive plans into a single type of incentive could reduce the size of the search space considerably. The amount of the reduction will be determined during attribute selection, when Predict will calculate the information gain for each attribute, and decide whether or not to keep the single attributes or the groups, or neither.

2.7 Summation of Number of Plans

Another transformation to improve the accuracy of the model could include the total number of incentive plans implemented. This could be a metric of the amount of dedication an employer or worksite has to their trip reduction program. For example if a worksite has 8 types of incentives, and another worksite only has 6, then the first worksite may seem more committed to their program. Incentives are not uniform in their costs and effectiveness, so this piece of information may be trivial.

2.8 Training and Testing Sets

In order to accurately evaluate the neural network models built; the models must be tested with unseen data. The data is separated into two disjoint sets, a training set and a testing set. There is a delicate balance on the sizes. If the testing set is too large, then there might not be enough data to accurately train the models. If the training set is too large, then there may not be enough data to accurately validate the models. Another option is k-fold cross-validation to show statistical significance. Cross-validation is especially useful in data sets with a small number of samples [1]. The size of the testing set is large (roughly 2100 samples); therefore cross-validation sets will not be created.

2.9 Final State of the Data

One observation that is immediately apparent is that that changes in VTR are small; therefore a reasonably accurate model could always predict the average change in VTR over the dataset. With regards to VTR, the smaller changes are not as important as the larger changes, because great increases or decreases are mostly the point of interest, because these plans show what to encourage when building a plan, and what to avoid. Figure 2. 1 shows the distribution of VTR for the Los Angeles data set. Since the target VTR is around 67, this figure shows that most plans fail to meet the goals set by the regulatory agency.

Table 2. 2 Los Angeles Incentives

Code	Description	Type of Incentive
BFL	Passenger Loading Areas	Facility Improvements
BFO	Other Facility Improvements	
BFP	Preferential Parking Areas	
BFR	Bike Racks and Bike Lockers	
BFS	Showers and Lockers	
BGA	TMA/TMO Guaranteed Ride Home	Guaranteed Ride Home
BGC	Company Vehicle Guaranteed Ride	
BGE	Emergency Ride Home	
BGO	Other Guaranteed Ride Home	
BGR	Rental Car Guaranteed Ride Home	
BGT	Taxi Guaranteed Ride Home	
BGU	Unscheduled O/T Ride Home	
BHF	Flextime for Ridesharers	Flextime
BHG	Flextime for Ridesharers	
BMC	Management Commitment	Marketing
BMF	Commuter Fairs	
BMG	Focus Groups	
BMM	Posted Materials	
BMN	New Hire Orientation	
BMO	Other Marketing Events	
BMP	Personal Communication	
BMR	Company Recognition	
BMS	Special Interest Club	
BMT	TMA/TMO Membership	
BMW	Written Materials	
BMZ	Promotional Meetings/Events	
BRC	Regional Commuter Management Agency	Rideshare Matching
BRE	Employer Rideshare Maching System	
DA	Transportation Allowances	
DFB	Bike to Work Subsidies	Direct Financial
DFC	Carpooling Subsidies	
FDFI	Transit Passes	
DFO	Other Direct Financial Subsidies	
DFS	Subsidized Vanpool Seats	
DFT	Transit Subsidies	
DFV	Vanpooling Subsidies	
DFW	Walk to Work Subsidies	
DNA	Auto Services	Direct Non-Financial
DNC	Gift Certificates	
DNF	Free Meals	
DNO	Other Direct Non-Financial Subsidies	

Table 2.2 (Continued)

Code	Description	Type of Incentive
DNP	Catalog Points	Direct Non-Financial
DNT	Additional Time Off With Pay	
DTH	Work at Home	Telecommuting
DTS	Work at Satellite Center	
DW3	3/36 Compressed Work Week	Compressed Work Week
DW4	4/40 Compressed Work Week	
DW9	9/80 Compressed Work Week	
DWO	Other Compressed Work Week	
IBO	Other Employee Benefits	
IBP	Drawings for Free Meals/Certificates/etc.	
IBV	Company Owned/Leased Vanpools	
ISC	Onsite Childcare	Onsite Services
ISO	Other Onsite Services	
ISS	Onsite Cafeteria/ATMs/Post Office	
IST	Onsite Transit Information or Pass Sales	
OOO	Other Not Classified by Other Codes	Other
XXX	Incentives Not Required	

Chapter 3

Artificial Neural Networks

3.1 Brief Introduction to Artificial Neural Networks

An Artificial Neural Network (ANN) is a set of computational units arranged in layers, which are interconnected. Each connection has a weight which changes the function represented by the network. Artificial neural networks are trained by adjusting the weights on the connections between artificial neurons [7].

Artificial Neural Networks (ANNs) were originally designed to be a loose interpretation of 1950s human cognition. Though it is improbable that any ANN would ever be able to accurately recreate the functionality of the human brain, these structures still prove useful. In this project, artificial neural networks will be used as a means of non-linear statistical modeling. This is much the same way that linear regression is used to build models from historical data. ANNs are useful in finding subtle complex relationships in data sets that may not be obvious to the human observer. The two types of ANNs to be discussed in this thesis are multi-layer perceptrons, and Radial Basis Function Neural Networks.

3.2 Radial Basis Function Networks

Radial Basis Function networks generally only have three layers. The first layer is the input layer, and serves the same purpose as in multi-layer networks. This input layer simply forwards the input values into the hidden units [5].

The second layer consists of the radial basis function neurons. These functions use a Gaussian activation function, as opposed to a sigmoid activation function. The neurons in the radial basis layer all have a vector called a center attached to them. The output of a neuron in the hidden layer is typically computed by taking the Euclidean distance from the center to the input vector x, and inputting the distance into a Gaussian function. The typical Gaussian function is shown in Figure 3. 1, and shown graphically in Figure 3. 2 with a sigma value of 5. In this equation the vector x is the feature vector, and the vector u is the vector that represents the center. The Gaussian function is a symmetric function, since |u-x| represents a distance value only positive values are shown in the graph.

The third layer consists of a single perceptron for each output value, which connects to the output. This perceptron is much like the perceptrons in the networks built by Predict. In this case there is only one perceptron in the third layer, because there is only a single output value (ΔVTR). The main difference between this perceptron and the perceptrons used in predict is that it uses a linear activation function, as opposed to a sigmoid function [2]. The linear activation function is shown in Figure 3. 3.

Figure 3. 1 Typical Gaussian Function

$$\varphi(r) = e^{\frac{-|u-x|^2}{2\sigma^2}}$$

Figure 3. 2 Graph of Typical Gaussian Function (Sigma=.5)

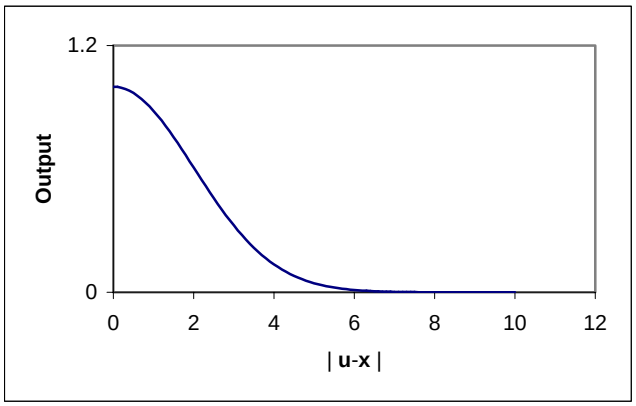


Figure 3. 3 Linear Activation Function

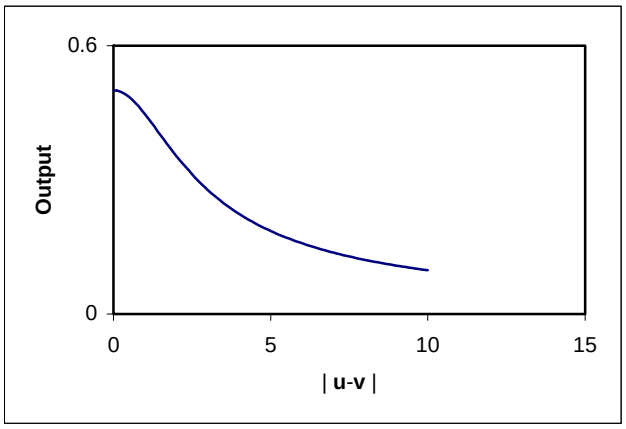
$$y_i = \sum w_i h_i$$

Other functions that could be used include inverse quadratics, such as the equation shown in Figure 3. 4 [2]. A graph of this function where c equals 2 is shown in Figure 3. 5. This function is similar to the Gaussian function, because it is a strictly decreasing function between 0 and 1. Any non-linear, strictly decreasing function that returns results between 1 and 0 could theoretically be used as activation functions for the radial basis. This is also a symmetric function, and |u-x| represents a distance value, therefore positive values are shown in Figure 3. 5.

Figure 3. 4 Typical Inverse Quadratic Function

$$\varphi(r) = \frac{1}{(|\mathbf{u}-\mathbf{x}|^2 + c^2)}$$

Figure 3. 5 Graph of Inverse Quadratic Function (c = 2)



3.2.1 Cover's Theorem

Using Radial Basis Function networks to approximate problems, which are not linearly separable, is rationalized by Cover's theorem. Cover's theorem on the separability of patterns states that a complex pattern cast in a high-dimensional space is more likely to be linearly separable than in a low dimensional space. In Radial Basis Function networks, the hidden units cast the problem into a high dimensional problem space with the non-linear hidden units [2].

3.2.2 Center Selection

Center selection is one of the most crucial aspects of building a radial basis function network. There are three commonly used methods of center selection [4][5]:

- Use all of the training inputs as centers, with a small σ value. This method of center selection works perfect on the training set, however this method does not work well on unseen data.
- Choose the initial centers randomly from the training set, and then adjust the centers during training.
- Use k-means clustering to choose representative centers that are not members of the training set.

3.3 Training of ANNs

Neural networks are traditionally trained by gradient descent methods such as backpropagation, or the Delta Rule[2]. These training methods are unlike methods such as simulated annealing, and only allow changes that reduce error, and are very prone to getting stuck in local minimum.

3.3.1 Training of Multi-Layer Perceptrons

Predict uses the cascade correlation training method proposed by Fahlman [3] to train and build artificial neural networks. This method starts with a minimal sized network and initially adjusts the weights, and then iteratively adds more neurons to attempt to improve the network [3].

3.3.2 Training of RBF Neural Networks

Radial basis neural networks work on the principle of locality, and the basic intuition that examples with similar attributes usually are more alike than examples with dissimilar attributes. Traditional RBF networks have a constant σ for each RBF neuron. This σ is similar to the learning constant, and is based on the experience of the network builder. This thesis will describe methods of training where the value of σ is adjusted during training, and may not be uniform for all neurons in the network. Only a change that positively affects the network is kept, if a change negatively affects the network the σ value is reverted to its previous value before the change. The following are methods that will be used in this thesis:

- The first method involves training the output layer for one epoch, and then adjusting the σ value for each RBF neuron between output layer training epochs.
- The second method involves training the output layer through all required epochs or until the minimum training gradient is reached and, then adjusts the σ value for each RBF until a minimum training gradient is reached. Minimum training

gradient refers to the smallest amount of improvement required for the algorithm to continue training.

The output layer (linear perceptron) will be trained using the Delta Rule training method for perceptron [2]. These adjusted methods of neural network training should provide a more accurate network.

3.4 Implementations

All multi-layer perceptron networks were created using a program called Predict. This program is implemented as a Microsoft Excel plug-in, which gets its data from an Excel worksheet. This software automates all phases of model building, and even can output "Flash Code" in Java and Visual Basic. This "Flash Code" can be easily dropped into an application with certain restrictions on licensing.

The radial basis neural network application written in C++ is a custom developed neural network-building program. This system implements two "hill-climbing" methods, the first is the Delta Rule for training the output layer, and the second is the dynamic center sizing for training the hidden layer.

Chapter 4

Building Neural Networks

4.1 Multi-Layer Perceptron Networks

The multi-layer perceptron networks were trained using Predict. Predict is a neural network training application, which works as a plug-in for Microsoft Excel. This application automates building neural networks using a training method known as cascade correlation [3].

The version of Predict used for building these networks was 3.0. This new version contains many enhancements over previous versions of Predict. It has added support of larger datasets; previous versions of predict only supported datasets with less than 16K records. Since the dataset contained over 25,000 records, and it was essential for this version to support the larger dataset.

Predict self evaluates the networks built by creating its own testing set from a subset of the input data. The size of this testing set and the method of selecting this set of data can be changed as an option. This option was kept at the default setting of 10% testing, chosen randomly.

Most of the Predict networks took approximately 10 hours to build on a computer with the attributes listed in Table 4.1. An experiment using the "exhaustive network search" parameter was used, and the network took 26 hours to build. After evaluating the network, the exhaustive network did not improve in performance over the comprehensive network. The major difference in the multi-layer networks shown in Table 4.2; is the depth of variable selection, depth of network search, and the amount of data

transformation (all other features of Predict were left at their default values). Fifteen different multi-layer networks were built and the results will be presented in section 4.1.3.

Table 4. 1 Attributes of PC Used to Build Multi-Layer Perceptron Networks

Processor	Pentium 3 1GHz
Main Memory	512 MB of PC133 SDRAM
Chipset	Intel 440BX
Operating System	Windows 2000 Professional
Manufactured by	Dell Computer Corporation

4.1.1 Data

The data consisted of 25,460 records, and 10% were randomly removed for use in evaluating the networks built. The testing set was removed from the training set by generating a uniformly distributed number between 0 and 1 for each record in the database. Records with a random number less than or equal to .10 were assigned into the testing set, and the other records were assigned to the training set. This process created a testing set of 2,537 records.

4.1.2 Training Issues

The greatest issue with Predict is that it is effectively a black box with dials, in which the data is fed in, and the network is fed out. Predict has these qualitative measures for certain parameters, such as “moderate”, “superficial”, “extensive”, and “comprehensive”. The heuristics behind these methods were not readily available.

During training, the network search parameter definitely had an effect on how long it took to train, however networks created using “comprehensive network search” as the network search parameter did not perform any better than the networks created using

the “moderate network search” option. Even though the moderate search took a matter of minutes, and the comprehensive search took overnight (14-16 hours).

One parameter that seemed very effective was the variable selection parameter. This feature uses statistical methods to determine which attributes have an impact on the target value. In a few instances, this feature took 95 attributes and reduced them down to 5 attributes. The variable selection parameter has a base value of “scale data only”, and an extreme value of “comprehensive variable selection”. Variable selection was useful, because it made the hypothesis space smaller, and therefore made the network search problem less difficult.

4.1.3 Multi-Layer Network Results

The results from the different multi-layer networks built using Predict are shown in Figure 4.2. The evaluations of these networks were made by placing the outputs in bins for each acceptable range as described in Section 2.4. Exact performance is defined as the percentage of records placed in the correct bin. Acceptable performance is defined as the percentage of records placed in the correct bin or one of its adjacent bins.

The networks built with Predict had very small architectures, therefore appeared to be following the principle of Occam’s Razor [9]. These networks were on the order of 20 neurons each. The MLP network that had the best performance had 31 input neurons, 18 hidden neurons and 1 output neuron.

Table 4. 2 Multi-Layer Network Results

	Exact Testing	Exact Training	Acceptable Testing	Acceptable Training
Network 1	19.35%	19.26%	42.18%	42.92%
Network 2	21.92%	20.37%	43.16%	43.32%
Network 3	19.98%	20.92%	43.32%%	44.48%
Network 4	19.75%	19.1%	42.14%	43.03%
Network 5	24.14%	20.27%	43.23%	43.25%
Network 6	20.5%	20.98%	42.69%	44.23%
Network 7	22.66%	19.1%	43.36%	42.8%
Network 8	20.58%	20.37%	42.57%	43.32%
Network 9	20.93%	20.29%	43.36%	43.94%
Network 10	18.53%	18.75%	42.22%	42.64%
Network 11	20.22%	20.44%	42.89%	43.36%
Network 12	19.55%	19.41%	42.14%	42.71%
Network 13	18.21%	18.41%	40.72%	41.5%
Network 14	18.64%	19%	41.03%	41.84%
Network 15	18.88%	19.39%	42.29%	43.66%
AVERAGE	20.26%	19.74%	42.49%	43.13%
STD DEV	1.64	0.82	0.81	0.8

The attributes in Table 4.3 show the attributes picked for the best performing feed forward network. There were 31 attributes chosen, these included a mix of mode split attributes and incentive attributes.

Table 4. 3 Attributes Selected For Multi-Layer Networks

Motorcycle Mode	Posted Marketing Materials
Single Occupant Vehicle Mode	New Hire Orientation
Two Person Carpool Mode	Other Marketing
Three Person Carpool Mode	Special Interest Club
Five Person Carpool Mode	Carpooling Subsidies
Telecommute Mode	Additional Time Off With Pay
Bicycle Mode	Increased Parking Costs for SOV
Compressed 3/36 Work Week Mode	Other Parking Management Strategies
Compressed 4/40 Work Week Mode	4/40 Incentive
Compressed 9/80 Work Week Mode	Compressed Work Week Incentive
Target AVR	Onsite Conveniences
Facility Improvements	Ride Home Incentives
Company Vehicle Guaranteed Ride	Parking Management Incentives
Emergency Guaranteed Ride	Other Compressed Work Week Incentive
Other Guaranteed Ride	VTR
Taxi Guaranteed Ride	

The accuracy for exact classification on each of the bins for the various multi-layer perceptron networks is shown in Figures 4.1 and 4.2. The distributions among the different bins are not very uniform, with a large spike in bin 3 and near zero in bins one and two. Since bins zero and one represent the desired case, zero percent accuracy in these two bins shows a serious flaw with this type of network.

Figure 4.1 Percent Accuracy by Bin of Testing Set (Multi-Layer)

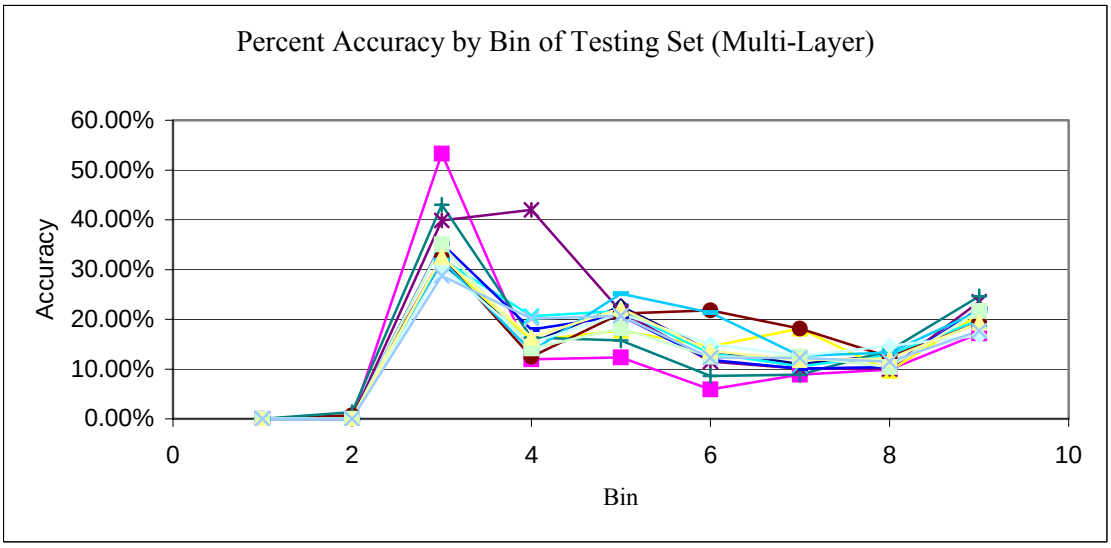
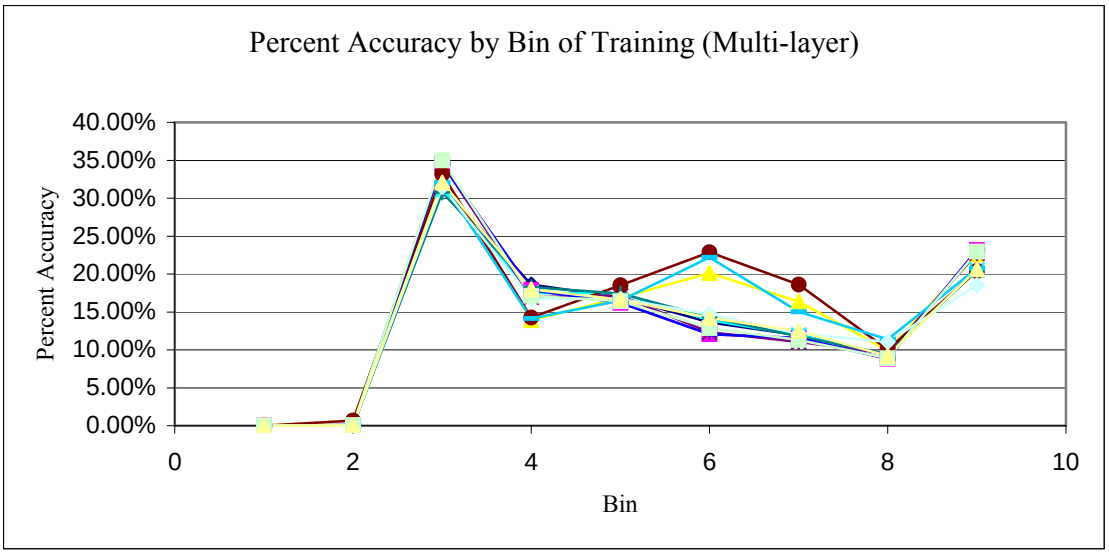


Figure 4 2 Percent Accuracy by Bin of Training Set (Multi-Layer)



4.2 Radial Basis Function Networks

The Radial Basis Function Networks were trained using a custom application developed specifically for this thesis. This application takes as input the training

attributes, training outputs, and centers. This system outputs a network trained as described in Chapter 3.

Since training of the output unit is analogous to training the simplest neural network, the training function was completed relatively quickly when compared to the back propagation networks. These networks were trained in a matter of an hour or two. This was dependent on the number of centers, because the output calculation for Gaussian units is more complex than the calculation of the linear output units.

One goal was to evaluate the k-means/dynamic width method versus the traditional method of training RBF neural networks. To evaluate traditional RBF networks, networks with 5000 centers chosen from the training set and were used to train these networks.

The radial basis networks were built on a machine with the attributes listed in Table 4.3. The k-means center based networks with 60 centers took about 2 hours to build, when the training gradient was set to 10^{-9}. The traditional networks with 5000 centers took about 12 hours to build with a minimum training gradient of 10^{-9}.

Table 4. 4 Attributes of PC Used to Build RBF Networks

Processor	AMD Athlon XP 2100+ (1.73 GHz)
Main Memory	512 MB PC 2700 (333 MHz DDR)
Chipset	VIA KT333
Operating System	Red Hat Linux v9.0
Manufactured by	Custom Built

4.2.1 Network Architectures

The network architectures for the RBF function networks had only three layers; the input layer, a single layer of hidden neurons, and the output layer. For this training

the number of hidden neurons were kept between three and one hundred, to prevent over fitting to the training set.

4.2.2 Center Selection

Centers were chosen using two different methods; the primary method is k-means clustering [6][8], and the secondary method was random sampling. The k-means centers were created by separating the training data into the three target areas; rural, suburban, and urban. These three data files were then clustered separately using the k-means clustering algorithm, each data file provided one-third of the total amount of centers. For example, a set of 60 centers would have 20 centers representing each target area. The centroids were then concatenated into a single file in order to build the center files.

The k-means algorithm makes a deterministic number of clusters indicated by a command line argument. The stopping criterion for the k-means clustering algorithm was 3000 iterations, or convergence which ever came first. Table 4.5 shows the different RBF layer sizes used in training the k-means networks.

4.2.3 Training Issues

One major issue in training was that if the training algorithm were initialized with too high a sigma value, the algorithm would not converge. For training, the sigma values were started between one and twenty for k-means based networks.

For networks with few centers this highest sigma value is a very large number, however networks with a large amount of centers (such as 150), this number becomes small. A table of highest sigma values is shown in Table 4.4. Since sigma sizes are adjusted in the negative direction, this does not become a problem during training, given an initial sigma value, which is below the highest sigma value threshold.

Table 4. 5 Limits of Sigma Value for K-means Centers

Number of RBF Centers	Highest sigma value
3	>>200
6	76
15	30
45	17
60	16
75	11
150	7

Another issue noted during training was that some companies do not implement any incentives at all. This may explain why the majority of ΔVTR values are close to zero. These records were not parsed out of the data, however after training roughly 27% of the records will have the same value, which is roughly the average of all ΔVTR values for records with no incentives implemented.

4.2.4 RBF Network Results

The RBF networks built were evaluated using the same method as the multi-layer networks as described in Section 4.1.3. Figures 4.3 through 4.6 compare the k-means center based RBF networks with dynamic centers against the best performing multi-layer network.

All attributes listed in Table 2.2 are used to build these networks. These attributes however are scaled. Mode split values are shown as percentages of the total mode split. Current VTR values are scaled from 0 to 1, with a VTR of 100 being 1 and a VTR of 0 remaining unchanged. Incentive values are already scaled between 0 and 1 and therefore are unchanged.

4.2.4.1 K-means Center Based

The k-means RBF network that was built using 60 centers exhibited the best performance, as shown in Figure 4.7. Networks of this type were built using 3, 6, 15, 45, 60, 75 and 150 centers, evenly distributed among the target areas described in Section 4.2.2. Therefore the performance of this network will be used to evaluate against the other types of networks.

The first comparison is on the exact classification for the testing set. A graph of the performance is shown in Figure 4.3. The line shown is the best performance of the Predict networks built. The RBF network starts to perform better than the best Predict network built at an initial sigma of 11.

Figure 4. 3 Exact Performance on the Testing Set (60 Centers)

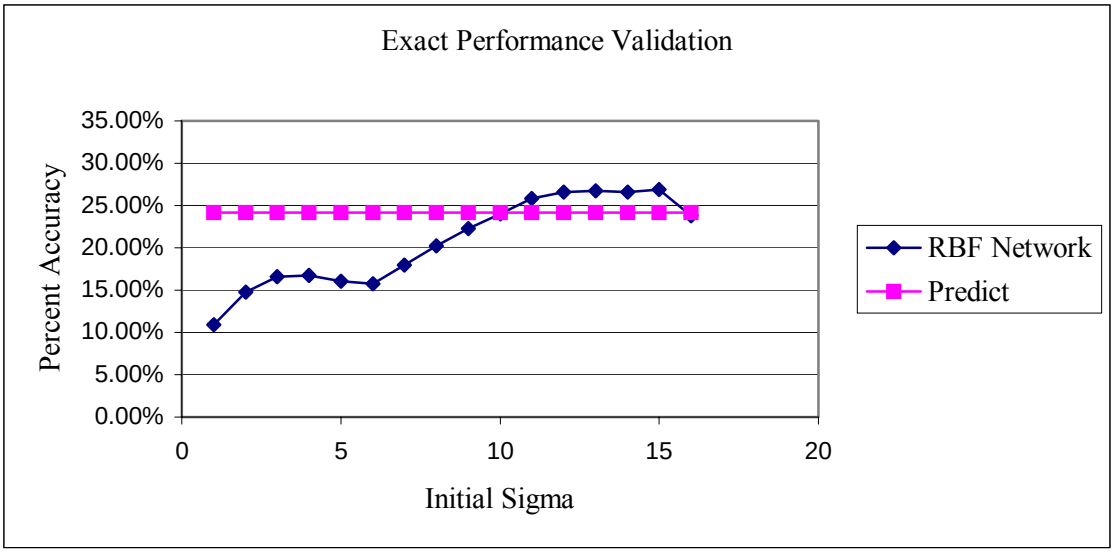
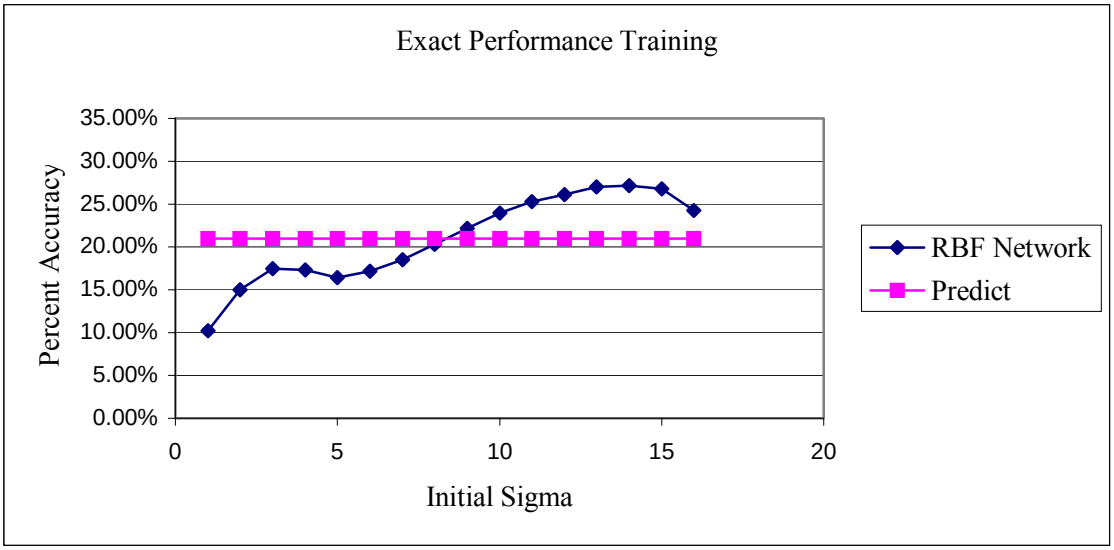


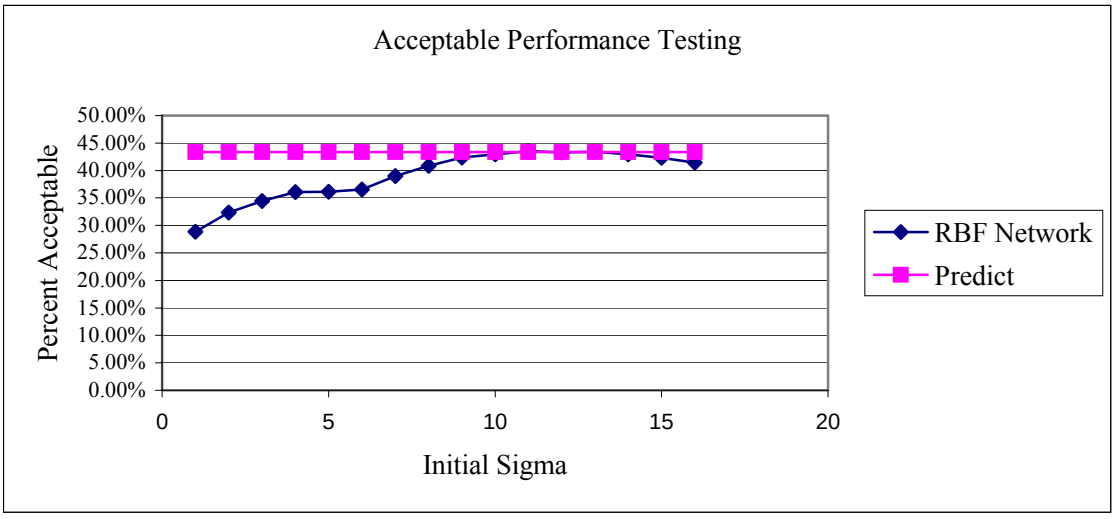
Figure 4. 4 Exact Performance on the Training Set (60 Centers)



The graph in Figure 4.4 shows the exact performance of the RBF network on the training set. As with Figure 4.1, the straight line is the best performance of all the Predict networks built at 20.92 percent. The radial basis network starts to perform better than the best Predict network at an initial sigma of roughly 9.5.

The next comparison is on acceptable performance for the testing set and is shown in Figure 4.5. The straight line notes the best performance for Predict at 43.36% acceptable. The performance of the Radial Basis Network is roughly the same as the Predict network between an initial sigma of 11 and 15.

Figure 4. 5 Acceptable Performance on the Testing Set (60 Centers)



The next comparison is on acceptable performance for the training set. The straight line notes the best performance for Predict at 44.48% acceptable. Figure 4.6 shows the relative performance. The performance of the Radial Basis Network is roughly the same as the Predict network between an initial sigma of 12 and 15.

Figure 4.7 shows a comparison between networks built with 30, 60, and 150 k-means centers. Initially, all three of these curves are tightly coupled, but the network with 60 centers performs better than the rest of the networks shown in this graph. These curves are similar for the training set.

Figure 4.6 Acceptable Performance on the Training Set (60 Centers)

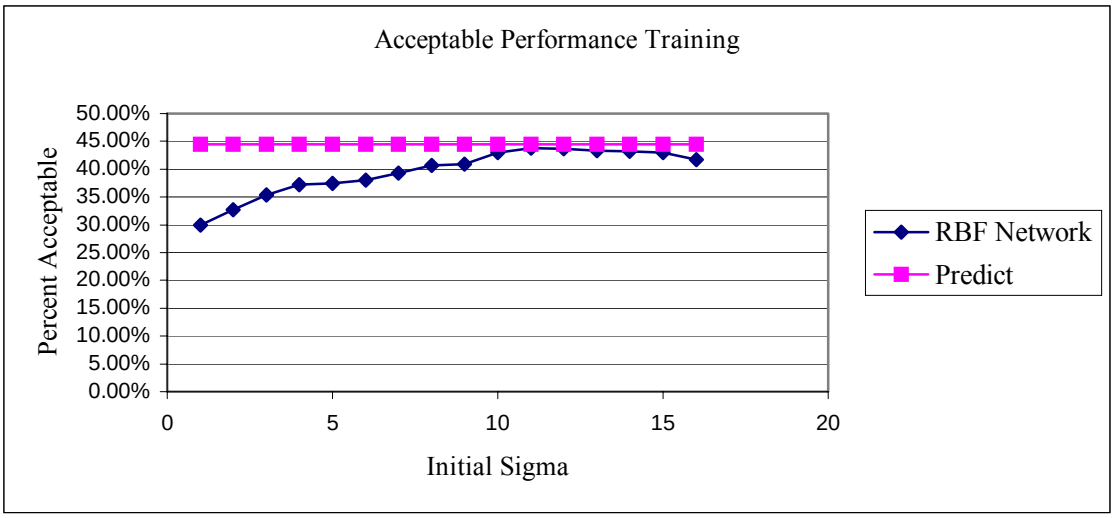


Figure 4.7 Exact Performance for K-means Centers on the Testing Set

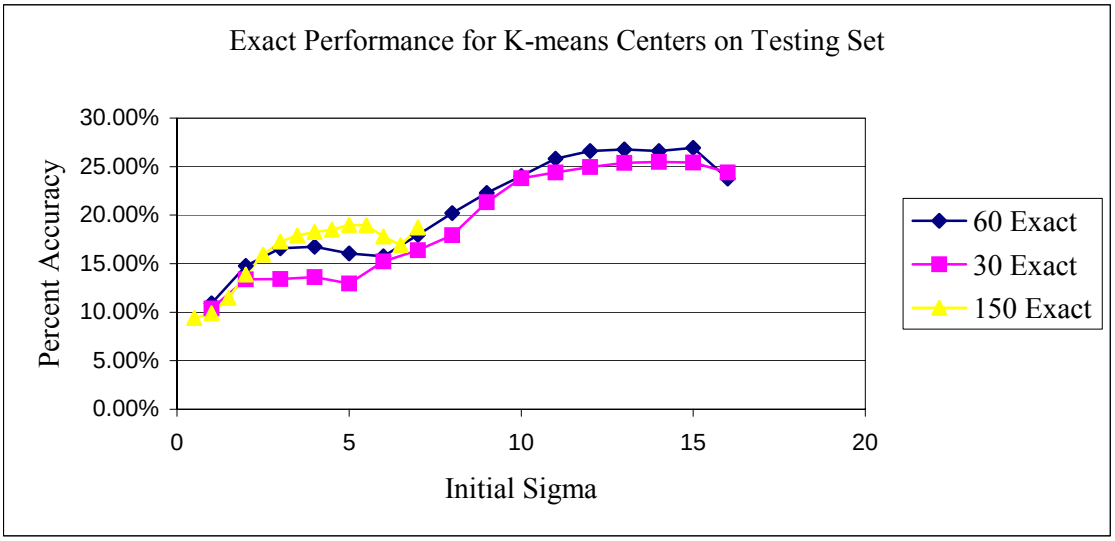
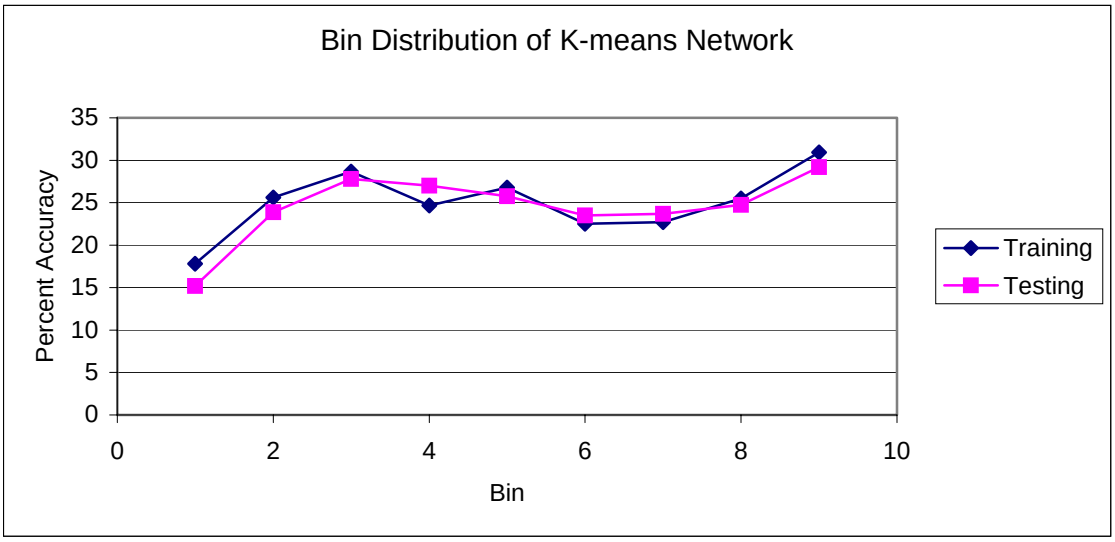


Figure 4.8 shows the exact classification accuracy of the k-means center based network for 60 centers across the different bins. When compared to Figures 4.1 and 4.2, the distribution appears to be much more uniform over the different bins. Also, when compared to Figures 4.1 and 4.2, the k-means network performs much better in bins one and two than the best multi-layer network.

Figure 4.8 Percent Accuracy Across Bins of K-means Network



4.2.4.2 Randomly Selected Center Based

The RBF networks trained with 5000 random centers are shown in Figures 4.9 and 4.10. The centers were chosen randomly from the training set and trained using the same method as the k-means centers based networks. For the exact performance classification, the network converged at around 17 percent accuracy. For the acceptable performance classification the network converged around 32 percent accuracy. These networks performed better than the k-means networks on the training set as expected for a small sigma value; however the performance on the testing set was much worse than on the multi-layer and k-means RBF networks.

Figure 4.9 Performance of 5000 Centers (Exact)

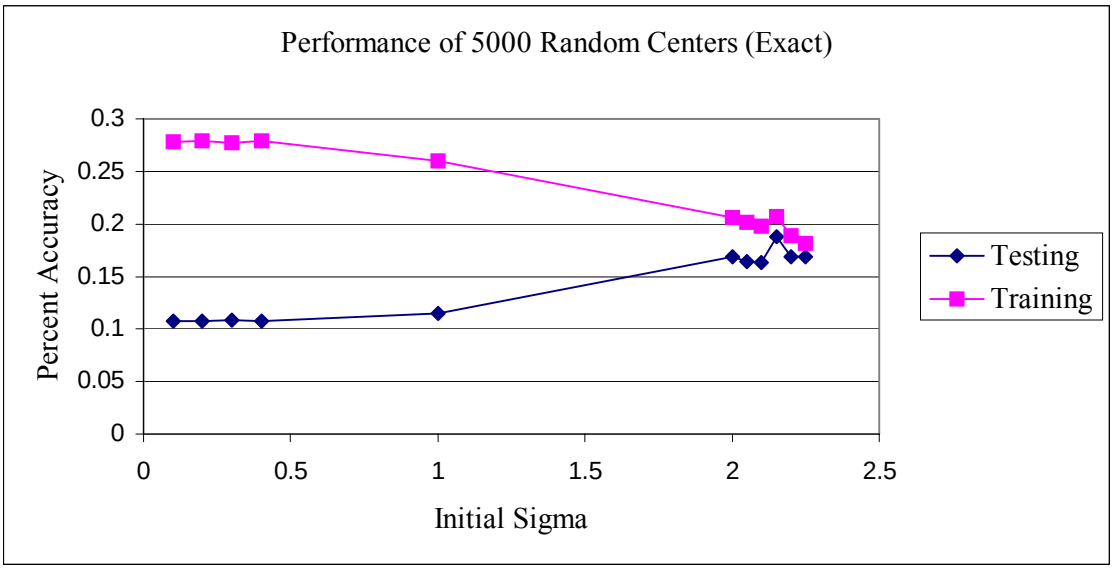
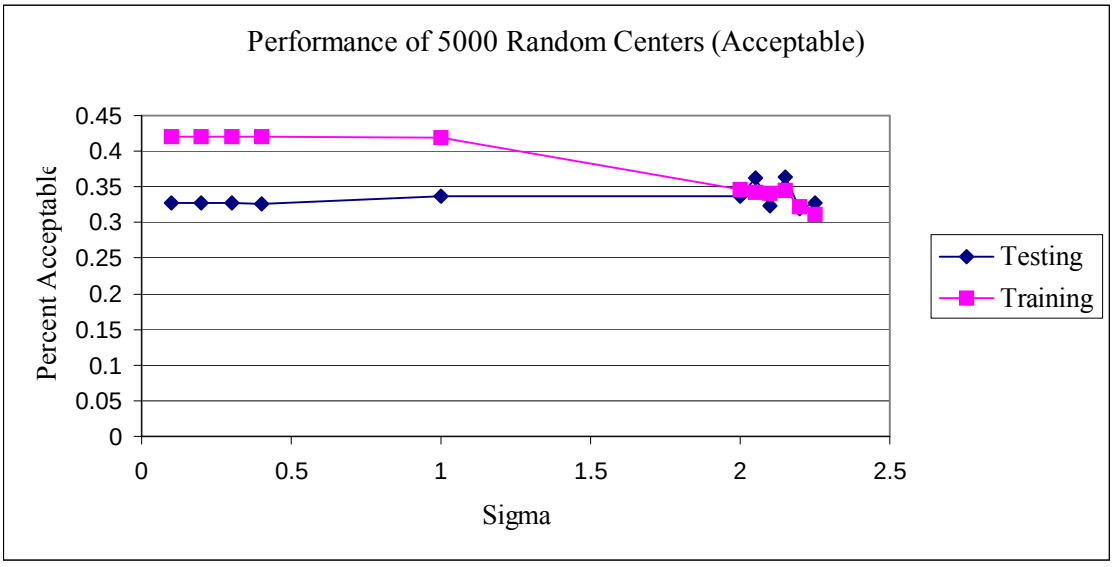
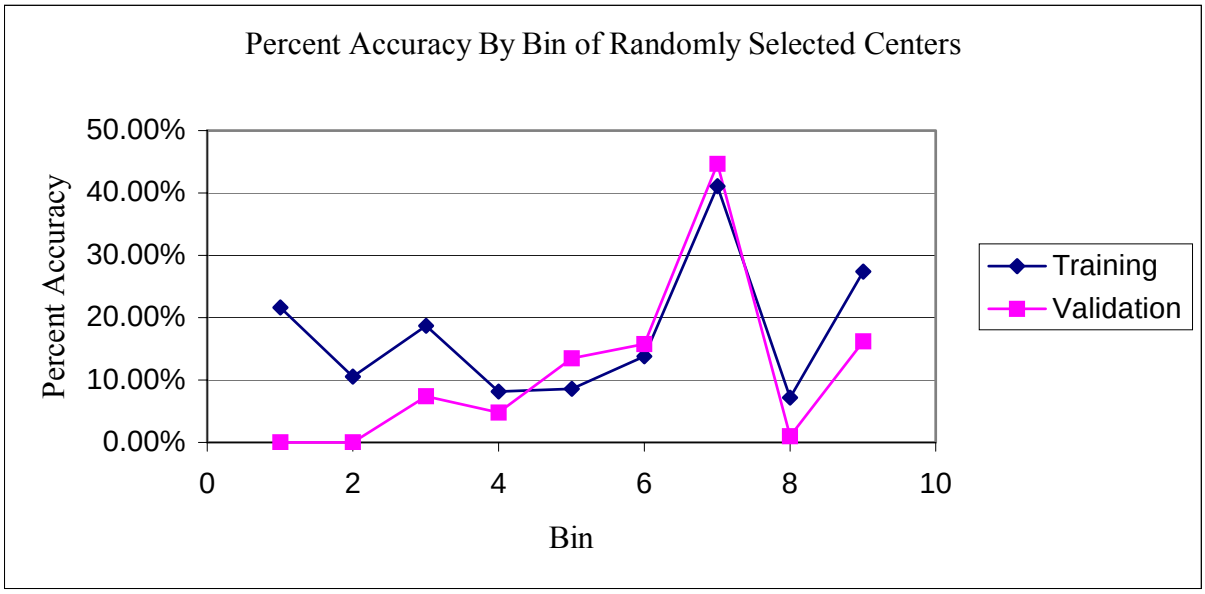


Figure 4.10 Acceptable performance of 5000 Centers (Acceptable)



The accuracy of this network across the different bins is shown in Figure 4.11. Like the multi-layer networks, this network exhibited near zero accuracy on bins one and two.

Figure 4.11 Percent Accuracy by Bin of Randomly Selected Centers



4.3 Comparison of Results

The radial basis networks built using the method described in this thesis have performed better than the multi-layer perceptron networks built by Predict. This is clearly shown in Figures 4.3 and 4.4. The radial basis networks built using the method described in this thesis have performed better than the classical RBF network. More importantly the accuracy across the different bins was relatively even for the network built with k-means based centers.

4.4 Generalization

The ability for a neural network to generalize to the problem space helps assure that this network reacts well to unseen data, and does not over fit to the training set [9]. The curves for the k-means center based RBF networks are similar for the training and testing set, thereby showing the power of building neural networks using this method. Figure 4.12 and 4.13 show the graphs of the training data, and testing data over the different sigmas. These Figures show that this method does not over fit to the training

data, no matter what the initial sigma value. This property is not held by the randomly chosen centers method, as clearly shown in Figures 4.9 and 4.10.

Figure 4.12 Comparison for Exact Classification

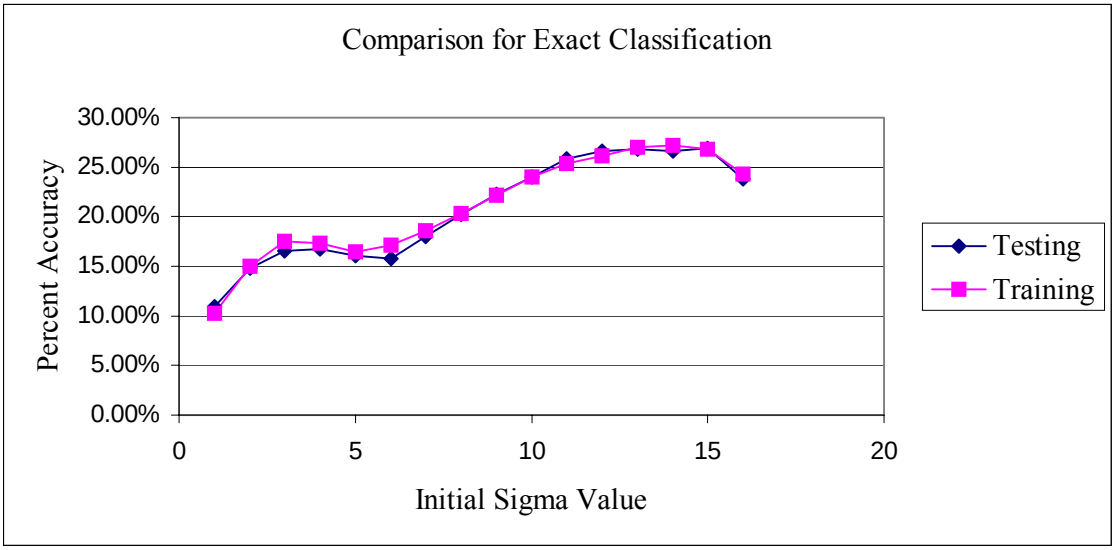
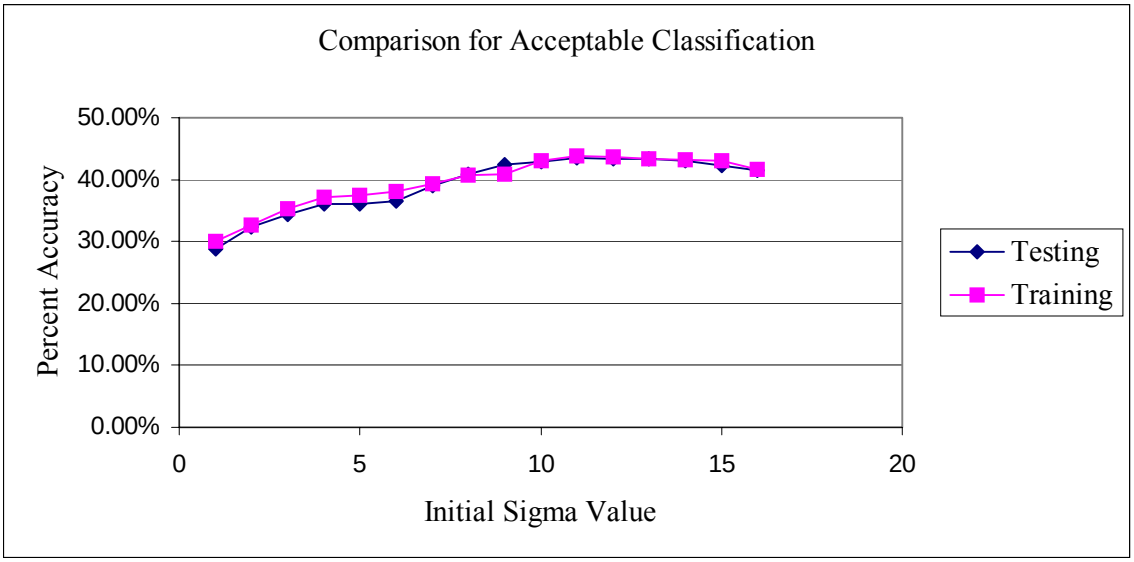


Figure 4.13 Comparison for Acceptable Classification



Chapter 5

Conclusion

The problem presented proved to be very difficult to model with any significant accuracy. However, the method presented in this thesis performed better than all methods tested for the following reasons:

- As shown in Section 4.4 this type of network seemed to generalize well. This is clearly illustrated in Figures 4.12 and 4.13. The training and testing performance curves are almost exactly the same, showing that for any sigma it generalizes well. This is not the case with networks built by randomly selected training data as shown in Figures 4.9 and 4.10.
- This type of network clearly performed much better than the network built with centers selected from the training set. The randomly selected centers based network had an accuracy of around 18 percent, while the k-means center based network had an accuracy of around 27 percent. This is an improvement of approximately 50 percent, which represents a great improvement.
- The k-means center based network performed marginally better than the multi-layer perceprton networks. The best performing multi-layer network had an accuracy of 24.14 percent, while the k-means center based networks had an accuracy of approximately 27 percent. This represents an improvement of around 12 percent. However this accuracy is spread more even across the different bins as clearly shown in Figures 4.1, 4.2 and 4.8.
- When compared to picking the most populated bin, the k-means center based networks performed approximately 4 percent better on exact classification. They performed approximately 10 percent better on the acceptable classification.

- The k-means center based networks performed better than random guess of the correct bin, which has an accuracy of around 11.1 percent.
- This type of network also performs better than guessing the most populated bin, which has an accuracy of around 22.5 percent. This represents an improvement of approximately 20 percent. This type of network also performs better than random guess in the acceptable category, which has an accuracy of 38 percent.
- One issue with the training method is that the selection of k for the k-means centering was able to see the testing set. This subtle advantage has the ability to skew the results towards the method presented, because the method of chosing the k-value has already seen the testing set.

Per bin accuracy is important, because a model could just always guess the common case and still get many records correct, and this model would not be sufficient, because all plans would have the same result. Therefore any changes made to the plan by the employer would not affect the output of the model.

These results also show that a few well-placed centers in an RBF network could outperform the brute force method of selecting many centers directly from the training set. The method of selecting from the training set only works well if the most of the problem space is represented by the training set. On the other hand, choosing centers using a clustering algorithm appears to perform as well on unseen data, as it does on the data on which the network was trained.

References

[1] T. Andersen, and T. Martinez, "Cross-validation and MLP architecture selection" *International Joint Conference on Neural Networks*, vol. 3, pp 1614-1619, 10-16 July 1999

[2] S. Haykin, *Neural Networks: A Comprehensive Foundation*

[3] Hoehfeld, and S.E. Fahlman, "Learning with Limited Numerical Precision Using the Cascade-Correlation Algorithm"*IEEE Transactions on Neural Networks*, vol 3, Issue: 4, pp 602 – 611, July 1992

[4] P. Panchapakesan and M. Palaniswami, "Effects of Moving the Centers in an RBF Network," *IEEE Transactions on Neural Networks*, vol. 13, pp. 1299-1307, 2002

[5] N, Karayiannis and G, Mi, "Growing Radial Basis Neural Networks: Merging Supervised and Unsupervised Learning with Network Growth Techniques," *IEEE Transactions on Neural Networks*, vol 8, pp 1492-1506, 1997

[6] H.S. Lee, and N.H. Younan, "An Investigation into Unsupervised Clustering Techniques" *Proceedings of the IEEE Southeastcon 2000*, pp 124-130, 7-9 April 2000

[7] Mizuno, et al., "Application of Neural Network To Technical Analysis of Stock Market Prediction", *Studies in Informatics and Control (With Emphasis on Useful Applications of Advanced Technology)*, June 1998.

[8] Z. Rong, and A.I. Rudnicky, "A Large Scale Clustering Scheme for Kernel K-Means"*16th International Conference on Pattern Recognition*, vol 4, pp 289-292, 2002

[9] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*

[10] P Winters, and S Hendricks, "Quantifying the Business Benefits of TDM," *http://www.dot.state.fl.us/research-center/Completed_Proj/Summary_PTO /FDOT_BC137_23_rpt.pdf*