Knowledge Discovery Using Neural Networks

Khosrow Kaikhah, Ph.D. and Sandesh Doddameti

Department of Computer Science
Texas State University
San Marcos, Texas 78666
kk02@TxState.edu
sd101017@TxState.edu

Abstract. A novel knowledge discovery technique using neural networks is presented. A neural network is trained to learn the correlations and relationships that exist in a dataset. The neural network is then pruned and modified to generalize the correlations and relationships. Finally, the neural network is used as a tool to discover all existing hidden trends in four different types of crimes in US cities as well as to predict trends based on existing knowledge inherent in the network.

1 Introduction

Enormous amounts of data are being generated and recorded for almost any kind of event or transaction. Advances in data storage and database technology have enabled us to store the vast amount of data very efficiently. A small piece of data may be quite insignificant. However, taken as a whole, data encompasses a vast amount of knowledge. A vital type of knowledge that can be acquired from large datasets are the hidden trends. These hidden trends, which can be expressed as rules or correlations, highlight the associations that exist in the data. For example, in a financial institution environment, where information about customer's characteristics and activities are maintained, the following trend may exist.

Persons who are between 25-30 years old, having at least a bachelor's degree with an income greater than 50K, have greater than 6 entertainment activities and greater than 10 restaurant activities in each cycle.

Finding these trends, which are specific to the application, represent a type of knowledge discovery. The acquired knowledge is helpful in understanding the domain, which the data describes.

We define a machine learning process that uses artificial neural networks to discover trends in large datasets. A neural network is trained to learn the inherent relationships among the data. The neural network is then modified

via pruning and hidden layer activation clustering. The modified neural network is then used as a tool to extract common trends that exist in the dataset as well as to predict trends. The extraction phase can be regulated through several control parameters.

2 Related Research

Andrews et al. in [1] discuss the difficulty in comprehending the internal process of how a neural network learns a hypothesis. According to their survey, rule extraction methods have been categorized into decompositional and pedagogical techniques. They discuss various techniques including the pros and cons of each method. The distinguishing characteristic of the decompositional approach is that the focus is on extracting rules at the level of individual (hidden and output) units within the trained Artificial Neural Network. In pedagogical approach to rule extraction, the trained neural network is treated as a black-box, in other words, the view of the underlying trained artificial neural network is opaque. They conclude that no single rule extraction/rule refinement technique is currently in a dominant position to the exclusion of all others.

Gupta et al. in [2] propose an algorithm (GLARE) to extract classification rules from feedforward and fully connected neural networks trained by backpropagation. The major characteristics of the GLARE algorithm are (a) its analytic approach for rule extraction, (b) its applicability to standard network structure and training method, and (c) its rule extraction mechanism as direct mapping between input and output neurons. This method is designed for a neural network with only one hidden layer. This approach uses the significance of connection strengths based on their absolute magnitude and uses only a few important connections (highest absolute values) to analyze the rules.

Our knowledge discovery process is both decompositional and pedagogical. It is decompositional in nature, since we examine the weights for pruning and clustering the hidden unit activation values. It is pedagogical, since we use the neural network as a black-box for knowledge discovery. Our approach is neither limited by the complexity of the hidden layer, nor by the number of hidden layers. Therefore our approach can be extended to networks with several hidden layers.

3 Our Approach

We have developed a novel process for discovering knowledge in datasets, with m dimensional input space and n dimensional output space, utilizing neural networks. Our process is independent of the application. The significance of our approach lies in using neural networks for discovering knowledge, with control parameters. The control parameters influence the discovery process in terms of importance and significance of the acquired knowledge. There are four phases in our approach: 1) neural network

training, 2) pruning and re-training, 3) clustering the hidden neuron activation values, and 4) rule discovery and extraction.

In phase one, the neural network is trained using a supervised learning method. The neural network learns the associations inherent in the dataset. In phase two, the neural network is pruned by removing all unnecessary connections and neurons. In phase three, the activation values of the hidden layer neurons are clustered using an adaptable clustering technique. In phase four, the modified neural network is used as a tool to extract and discover hidden trends. These four phases are described in more detail in the next four sections.

3.1 Neural Network Training

Neural networks are able to solve highly complex problems due to the non-linear processing capabilities of their neurons. In addition, the inherent modularity of the neural network's structure makes them adaptable to a wide range of applications [3]. The neural network adjusts its parameters to accurately model the distribution of the provided dataset. Therefore, exploring the use of neural networks for discovering correlations and trends in data is prudent.

The input and output patterns may be real-valued or binary-valued. If the patterns are real-valued, each value is discretized and represented as a sequence of binary values, where each binary value represents a range of real values. For example, in a credit card transaction application, an attribute may represent the person's age (a value greater than 21). This value can be discretized into 4 different intervals: (21-30],(30-45],(45-65], and (65+]. Therefore [0 1 0 0] would represent a customer between the ages of 31 and 45. The number of neurons in the input and output layers are determined by the application, while the number of neurons in the hidden layer are dependent on the number of neurons in the input and output layers.

We use an augmented gradient descent approach to train and update the connection strengths of the neural network. The gradient descent approach is an intelligent search for the global minima of the energy function. We use an energy function, which is a combination of an error function and a penalty function [4]. The error function computes the error of each neuron in the output layer, and the penalty function drives the connection strengths of unnecessary connections to very small values while strengthening the rest of the connections. The penalty function is defined as:

$$P(w,v) = \rho_{decay}\left(P_1(w,v) + P_2(w,v)\right) \tag{1}$$

$$P_1(w,v) = \varepsilon_1\left(\sum_{j=1}^{h}\sum_{i=1}^{m}\frac{\beta w_{ij}^2}{1+\beta w_{ij}^2} + \sum_{j=1}^{h}\sum_{k=1}^{n}\frac{\beta v_{jk}^2}{1+\beta v_{jk}^2}\right) \tag{1a}$$

$$P_2(w,v) = \varepsilon_2\left(\sum_{j=1}^{h}\sum_{i=1}^{m} w_{ij}^2 + \sum_{j=1}^{h}\sum_{k=1}^{n} v_{jk}^2\right) \qquad (1b)$$

The network is trained till it reaches a recall accuracy of 99% or higher.

3.2 Pruning and Re-Training

The neural network is trained with an energy function, which includes a penalty function. The penalty function drives the strengths of unnecessary connections to approach zero very quickly. Therefore, the connections having very small values, values less than 1, can safely be removed without significant impact on the performance of the network. After removing all weak connections, any input layer neuron having no emanating connections can be removed. In addition, any hidden layer neuron having no abutting or emanating connections can safely be removed. Finally, any output layer neuron having no abutting connections can be removed. Removal of input layer neurons correspond to having irrelevant inputs in the data model; removal of hidden layer neurons reduces the complexity of the network and the clustering phase; and removal of the output layer neurons corresponds to having irrelevant outputs in the data model. Pruning the neural network results in a less complex network while improving its generalization.
Once the pruning step is complete, the network is trained with the same dataset in phase one to ensure that the recall accuracy of the network has not diminished significantly. If the recall accuracy of the network drops by more than 2%, the pruned connections and neurons are restored and a stepwise approach is pursued. In the stepwise pruning approach, the weak incoming and outgoing connections of the hidden layer neurons are pruned, one neuron at a time, and the network is re-trained and tested for recall accuracy.

3.3 Clustering the Hidden Layer Neuron Activation Values

The activation values of each hidden layer neuron are dynamically clustered and re-clustered with a cluster radius and confidence radius, respectively. The clustering algorithm is adaptable, that is, the clusters are created dynamically as activation values are added into the clusterspace. Therefore, the number of clusters and the number of activation values in each cluster are not known *a priori*. The centroid of each cluster represents the mean of the activation values in the cluster and can be used as the representative value of the cluster, while the frequency of each cluster represents the number of activation values in that cluster. By using the centroids of the clusters, each hidden layer neuron has a minimal set of activations. This helps with getting generalized outputs at the output layer. The centroid of a cluster c is denoted by G_c. The centroid is adjusted dynamically as new elements e_c^i are added to the cluster.

$$G_c^{new} = \frac{(G_c^{old} \cdot freq_c) + e_c^i}{freq_c + 1} \quad (2)$$

Since dynamic clustering is order sensitive, once the clusters are dynamically created with a cluster radius that is less than a predetermined upper bound, all elements will be re-clustered with a confidence radius of one-half the cluster radius. The upper bound for cluster radius defines a range for which the hidden layer neuron activation values can fluctuate without compromising the network performance.

The benefits of re-clustering are twofold: 1) Due to order sensitivity of dynamic clustering, some of the activation values may be misclassified. Re-clustering alleviates this deficiency by classifying the activation values in appropriate clusters. 2) Re-clustering with a different radius (confidence radius) eliminates any possible overlaps among clusters. In addition during re-clustering, the frequency of each confidence cluster is calculated, which will be utilized in the extraction phase.

3.4 Knowledge Discovery

In the final phase of the process, the knowledge acquired by the trained and modified neural network is extracted in the form of rules [5], [6]. This is done by utilizing the generalization of the hidden layer neuron activation values as well as control parameters. The novelty of the extraction process is the use of the hidden layer as a filter by performing vigilant tests on the clusters. Clusters identify common regions of activations along with the frequency of such activities. In addition, clusters provide representative values (the mean of the clusters) that can be used to retrieve generalized outputs.

The control parameters for the extraction process include: a) cluster radius, b) confidence frequency, and c) hidden layer activation level. The cluster radius determines the coarseness of the clusters. The confidence radius is usually set to one-half of the cluster radius to remove any possible overlaps among clusters. The confidence frequency defines the minimum acceptable rate of commonality among patterns. The hidden layer activation level defines the maximum level of tolerance for inactive hidden layer neurons.

Knowledge extraction is performed in two steps. First, the existing trends are discovered by presenting the input patterns in the dataset to the trained and modified neural network and by providing the desired control parameters. The input patterns that satisfy the rigorous extraction phase requirements and produce an output pattern represent generalization and correlations that exist in the dataset. The level of generalization and correlation acceptance is regulated by the control parameters. This ensures that inconsistent patterns, which fall outside confidence regions of hidden layer activations, or fall within regions with low levels of activity, are not considered. There may be many duplicates in these accepted input-output

pairs. In addition, several input-output pairs may have the same input pattern or the same output pattern. Those pairs having the same input patterns will be combined, and, those pairs having the same output patterns will be combined. This post-processing is necessary to determine the minimal set of trends. Any input or output attribute not included in the discovered trend corresponds to irrelevant attributes in the dataset. Second, the predicated trends are extracted by providing all possible permutations of input patterns, as well as the desired control parameters. Any additional trends discovered in this step constitute the predicated knowledge based on existing knowledge. This step is a direct byproduct of the generalizability of neural networks.

4 Discovering Trends in Crimes in US Cities

We compiled a dataset consisting of the latest annual demographic and crime statistics for 6100 US cities. The data is derived from three different sources: 1) US Census; 2) Uniform Crime Reports (UCR) published annually by the Federal Bureau of Investigation (FBI); 3) Unemployment Information from the Bureau of Labor Statistics.

We used the dataset to discover trends in crimes with respect to the demographic characteristics of the cities. We divided the dataset into three groups in terms of the population of cities: a) cities with populations of less than 20k (4706 cities), b) cities with populations of greater than 20k and less than 100k (1193 cities), and c) cities with populations of greater than 100k (201 cities). We then trained a neural network for each group and each of four types of crimes (murder, rape, robbery, and auto theft), a total of 12 networks. We divided the dataset into three groups in terms of city population, since otherwise, small cities (cities less that 20k) would dominate the process due to their high overall percentage. Table 1 includes the demographic characteristics and crime types we used for the knowledge discovery process.

Table 1: The Categories for the Process

I_1: *City Population*
I_2: *Percentage of Single-Parent Households*
I_3: *Percentage of Minority*
I_4: *Percentage of Young People(between the ages of 15 and 24)*
I_5: *Percentage of Home Owners*
I_6: *Percentage of People living in the Same House since 1985*
I_7: *Percentage of Unemployment*
O_1: *Number of Murders*
O_2: *Number of Rapes*
O_3: *Number of Robberies*
O_4: *Number of Auto Thefts*

Each category is discretized into several intervals to define the binary input/output patterns. For each crime type, three different neural networks are trained for the three groups of cities (small, medium, and large) to an accuracy of 99% or higher. Each network consists of 44 input layer neurons, 60 hidden layer neurons, and 4 to 5 output layer neurons. After the training phase, the networks are pruned and clustered. Although, for each network, about 30% of connections as well as about 5% of hidden layer neurons were pruned, none of the input neurons were pruned. This reflects the importance of all demographic categories we used for discovering trends in crimes. After phase two and three, all networks maintain an accuracy rate of 99% or higher. The networks were then used as tools to discover the existing, as well as predicted trends. Table 2 represents the discrete intervals for each category.

Table 2: Discrete Intervals

Categories	N	Intervals
I_1 (*small*)	5	[0-4k],(4k-8k],(8k,12k],(12k-16k],(16k-20]
I_1 (*medium*)	5	(20k-40k],(40k-60k],(60k-80k],(80k-90k],(90k-100k]
I_1 (*large*)	5	(100k-130k],(130k-160k],(160k-200k],(200k-500k],500k+
I_2	7	[0-5],(5-7],(7-9],(9-11],(11-14],(14-20],(20-100]
I_3	6	[0-5],(5-10],(10-20],(20-40],(40-70],(70-100]
I_4	7	[0-12],(12-13],(13-14],(14-15],(15-17],(17-25],(25-100]
I_5	7	[0-40],(40-50],(50-60],(60-70],(70-80],(80-90],(90-100]
I_6	6	[0-45],(45-50],(50-55],(55-60],(60-65],(65-100]
I_7	6	[0-4],(4-6],(6-8],(8-12],(12-20],(20-100]
O_1	4	0, (1-5],(5-10],10+
O_2	5	0, (1-5],(5-10],(10-70],70+
O_3	5	0, (1-5],(5-10],(10-100],100+
O_4	5	[0-10],(10-100],(100-500],(500-1000],1000+

4.1 Trends in Small Cities

The following are the existing trends discovered for small cities.

1) $(0<I_1\le 4k)\wedge(0<I_3\le 5)\wedge(0<I_7\le 4)\wedge(7<I_2\le 9)\wedge(50<I_6\le 55)\wedge(14<I_4\le 15)\wedge(70<I_5\le 80)\Rightarrow O_1=0$

2) $(0k<I_1\le 8k)\wedge(0<I_3\le 5)\wedge(0<I_7\le 4)\wedge[(0<I_2\le 5)\vee(7<I_2\le 9)]\wedge(0<I_6\le 50)\wedge(0<I_4\le 13)\wedge(60<I_5\le 70)\Rightarrow 1<O_3\le 5$

3) $(4k<I_1\le 8k)\wedge(0<I_3\le 5)\wedge(6<I_7\le 8)\wedge(9<I_2\le 11)\wedge(65<I_6\le 80)\wedge(13<I_4\le 14)\wedge(60<I_5\le 70)\Rightarrow 1<O_4\le 5$

4) $(8k<I_1\le 16k)\wedge(0<I_3\le 5)\wedge[(0<I_7\le 4)\vee(8<I_7\le 12)]\wedge(7<I_2\le 9)\wedge(55<I_6\le 60)\wedge(13<I_4\le 14)\wedge(60<I_5\le 70)\Rightarrow O_4=0$

The following are the predicted trends discovered for small cities.

1) $[(0<I_1\le 4k)\vee(12k<I_1\le 16k)]\wedge(0<I_3\le 5)\wedge(0<I_7\le 6)\wedge(0<I_2\le 5)\wedge$
$(0<I_6\le 55)\wedge(13<I_4\le 14)\wedge(70<I_5\le 80)\Rightarrow O_1=0$

2) $(0<I_1\le 4k)\wedge(20<I_3\le 40)\wedge(4<I_7\le 6)\wedge(9<I_2\le 11)\wedge(0<I_6\le 45)\wedge$
$(13<I_4\le 14)\wedge(50<I_5\le 60)\Rightarrow O_2=0$

3) $(4k<I_1\le 8k)\wedge(5<I_3\le 10)\wedge(4<I_7\le 6)\wedge(5<I_2\le 7)\wedge(60<I_6\le 65)\wedge$
$(15<I_4\le 17)\wedge(40<I_5\le 50)\Rightarrow 1<O_2\le 5$

4) $(12k<I_1\le 16k)\wedge(5<I_3\le 10)\wedge(4<I_7\le 6)\wedge(5<I_2\le 7)\wedge(0<I_6\le 45)\wedge$
$(17<I_4\le 25)\wedge(80<I_5\le 90)\Rightarrow 1<O_3\le 5$

4.2 Trends in Medium Cities

The following are the existing trends discovered for medium cities.

1) $(20k<I_1\le 40k)\wedge(0<I_3\le 5)\wedge(0<I_7\le 6)\wedge[(5<I_2\le 7)\vee(9<I_2\le 11)]\wedge$
$(45<I_6\le 55)\wedge(12<I_4\le 13)\wedge(60<I_5\le 90)\Rightarrow O_1=0$

2) $(20k<I_1\le 40k)\wedge(10<I_3\le 20)\wedge(0<I_7\le 4)\wedge(5<I_2\le 7)\wedge$
$(0<I_6\le 45)\wedge(12<I_4\le 13)\wedge(60<I_5\le 70)\Rightarrow 1<O_2\le 5$

3) $(20k<I_1\le 40k)\wedge(5<I_3\le 10)\wedge(4<I_7\le 6)\wedge(11<I_2\le 14)\wedge$
$(45<I_6\le 50)\wedge(14<I_4\le 15)\wedge(40<I_5\le 50)\Rightarrow 1<O_2\le 10$

4) $(20k<I_1\le 40k)\wedge(0<I_3\le 5)\wedge(4<I_7\le 6)\wedge(0<I_2\le 5)\wedge$
$(65<I_6\le 80)\wedge(12<I_4\le 13)\wedge(80<I_5\le 90)\Rightarrow 10<O_4\le 100$

5) $(20k<I_1\le 40k)\wedge(0<I_3\le 5)\wedge(4<I_7\le 6)\wedge(9<I_2\le 11)\wedge$
$(0<I_6\le 45)\wedge(14<I_4\le 15)\wedge(50<I_5\le 60)\Rightarrow 100<O_4\le 500$

The following are the predicted trends discovered for medium cities.

1) $(20<I_1\le 40k)\wedge(40<I_3\le 100)\wedge[(0<I_7\le 4)\vee(6<I_7\le 8)]\wedge(5<I_2\le 9)\wedge$
$(40<I_6\le 55)\wedge[(13<I_4\le 14)\vee(17<I_4\le 25)]\wedge(40<I_5\le 60)\Rightarrow 1<O_2\le 10$

2) $(60k<I_1\le 80k)\wedge(10<I_3\le 20)\wedge(4<I_7\le 6)\wedge(7<I_2\le 9)\wedge$
$(50<I_6\le 55)\wedge(15<I_4\le 17)\wedge(80<I_5\le 90)\Rightarrow O_3=0$

3) $(20k<I_1\le 40k)\wedge(70<I_3\le 100)\wedge(6<I_7\le 8)\wedge(5<I_2\le 7)\wedge$
$(0<I_6\le 45)\wedge(0<I_4\le 12)\wedge(80<I_5\le 90)\Rightarrow 10<O_3\le 100$

4.3 Trends in Large Cities

The following are the existing trends discovered for large cities.

1) $(200k < I_1 \le 500k) \wedge (20 < I_3 \le 40) \wedge (8 < I_7 \le 12) \wedge (11 < I_2 \le 14) \wedge$
$(50 < I_6 \le 55) \wedge (15 < I_4 \le 17) \wedge (50 < I_5 \le 60) \Rightarrow 5 < O_1 \le 10$

2) $(200k < I_1 \le 500k) \wedge (40 < I_3 \le 70) \wedge (8 < I_7 \le 12) \wedge (14 < I_2 \le 20) \wedge$
$(50 < I_6 \le 55) \wedge (15 < I_4 \le 17) \wedge (50 < I_5 \le 60) \Rightarrow 10 < O_2 \le 70$

3) $(160k < I_1 \le 200k) \wedge (20 < I_3 \le 40) \wedge (6 < I_7 \le 8) \wedge (11 < I_2 \le 14) \wedge$
$(45 < I_6 \le 50) \wedge (14 < I_4 \le 17) \wedge (50 < I_5 \le 60) \Rightarrow O_3 > 100$

4) $(100k < I_1 \le 130k) \wedge (10 < I_3 \le 20) \wedge (0 < I_7 \le 4) \wedge (5 < I_2 \le 7) \wedge$
$(0 < I_6 \le 45) \wedge (0 < I_4 \le 12) \wedge (70 < I_5 \le 80) \Rightarrow 500 < O_4 \le 1000$

The following are the predicted trends discovered for large cities.

1) $(200k < I_1 \le 500k) \wedge (70 < I_3 \le 100) \wedge (6 < I_7 \le 8) \wedge (11 < I_2 \le 14) \wedge$
$[(0 < I_6 \le 45) \vee (50 < I_6 \le 55)] \wedge [(13 < I_4 \le 14) \vee (17 < I_4 \le 25)] \Rightarrow O_2 > 70$

2) $(100k < I_1 \le 130k) \wedge (70 < I_3 \le 100) \wedge (6 < I_7 \le 8) \wedge (11 < I_2 \le 14) \wedge$
$[(45 < I_6 \le 50) \vee (60 < I_6 \le 65)] \wedge (17 < I_4 \le 25) \wedge (50 < I_5 \le 60) \Rightarrow O_3 > 100$

3) $(200k < I_1 \le 500k) \wedge (70 < I_3 \le 100) \wedge (8 < I_7 \le 12) \wedge (11 < I_2 \le 14) \wedge$
$(60 < I_6 \le 65) \wedge (0 < I_4 \le 12) \wedge (60 < I_5 \le 70) \Rightarrow O_4 > 1000$

4) $(100k < I_1 \le 130k) \wedge (5 < I_3 \le 10) \wedge (12 < I_7 \le 20) \wedge (9 < I_2 \le 11) \wedge$
$(45 < I_6 \le 50) \wedge (13 < I_4 \le 14) \wedge (0 < I_5 \le 40) \Rightarrow 100 < O_4 \le 1000$

5 Conclusions

For each group of cities (small, medium, large), we are able to discover the existing trends for each type of crime (murder, rape, robbery, auto theft). These trends represent the hidden knowledge and are based on the high level of commonalty inherent in the dataset. The desired level of commonalty can be regulated through the control parameters. In addition, by using the generalizability feature of neural networks, we are able to discover predicted trends. These trends describe the demographic characteristics of cities that contribute to each type of crime. Once again, the control parameters provide the ability to regulate the desired level of commonality. According to the experts in criminal fields, the discovered trends accurately reflect the reality that exists in US cities. They were particularly impressed with the predicted trends, since they can use this knowledge for restructuring their resources.

The knowledge discovery technique can be applied to any application domain that deals with vast amounts of data such as medical, military, business, and security. In medical fields, the data gathered from cancer

patients can be used to discover the dominating factors and trends for the development of cancer. In military fields, the data gathered from the enemy can be used to predicate their future movements. In business environments, the data gathered from customers can be used to model the transaction activities of the customers. In security applications, the data gathered can be used to predicate and prevent potential intrusions.

6 References

1. Robert Andrews, Joachim Diederich, and Alan Tickle, "A Survey and Critique of Techniques for Extracting Rules from Trained Artificial Neural Networks", *Neurocomputing Research Center*, 1995

2. Amit Gupta, Sang Park, and Siuva M. Lam, "Generalized Analytic Rule Extraction for Feedforward Neural Networks", *IEEE transactions on knowledge and data engineering*, 1999

3. Kishan Mehrotra, Chilukuri K. Mohan, and Sanjay Ranka, *Elements of Artificial Neural Networks (Complex Adaptive Systems)*, Cambridge, MA: MIT Press, 1997

4. Rudy Setiono, "Extracting Rules from Pruned Neural Networks for Breast Cancer Diagnosis", *Artificial Intelligence in Medicine*, 1996

5. Rudy Setiono and Huan Liu, "Effective Data Mining Using Neural Networks", *IEEE transactions on knowledge and data engineering*, 1996

6. Tony Kai, Yun Chan, Eng Chong Tan, and Neeraj Haralalka, "A Novel Neural Network for Data Mining", *8th International Conference on Neural Information Processing Proceedings. Vol.2*, 2001

7. Mark W. Craven and Jude W. Shavlik, "Using Neural Networks for Data Mining", *Future Generation Computer Systems special issue on Data Mining*, 1998

8. Jason T. L. Wang, Qicheng Ma, Dennis Shasha, and Cathy H.Wu, kkk"Application of Neural Networks to Biological Data Mining: A Case Study in Protein Sequence Classification", *The Sixth ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, August 20-23, 2000 Boston, MA, USA.

9. Joseph P. Bigus, *Data Mining With Neural Networks: Solving Business Problems from Application Development to Decision Support*, McGraw-Hill, NY, 1996