A Comparative Study of Different Machine Learning Approaches for Decision Making

ARPAD KELEMEN[1], YULAN LIANG[2], STAN FRANKLIN[1]
[1]The Institute for Intelligent Systems
The University of Memphis
Memphis TN 38152
USA
[2]Department of Mathematical Sciences
The University of Memphis
Memphis, TN 38152
USA
kelemena@msci.memphis.edu   http://www.msci.memphis.edu/~kelemena

Abstract: - In this paper, several neural network and statistical learning approaches are proposed that learn to make human like decisions for the job assignment problem of the US Navy. Comparison study of Feedforward Neural Networks (FFNN), Adaptive Neuro-Fuzzy Inference System (ANFIS), Support Vector Machine (SVM) and Adaptive Bayes (AB) classifier with Generalized Estimation Equation (GEE) is provided. Although Support Vector Machine provided the highest decision making rate, other methods also offer various unique advantages. In spite of the high correlation naturally present in the data, with the use of post-processing results were further improved.

Key-Words: - Feedforward Neural Network, Adaptive Neuro-Fuzzy Inference System, Support Vector Machine, Adaptive Bayes Classifier

1 Introduction

Every US Navy sailor has to be assigned to a new job from time to time according to mandatory Navy policy of sea/shore duty rotation. It is important that the right job is assigned to the right sailor to achieve sailor and Navy happiness. However, many, often conflicting, criteria complicate the assignment process. On one hand, jobs need to be assigned according to the ability of the sailor. On the other hand, the needs of the assignee, like reluctance to be posted to jobs far away from home for too long, have to be addressed also. Moreover, the process is complicated by other factors. Firstly, the various criteria, or constraints, can be soft, hard and semi-hard. Secondly, different navy experts (detailers) may provide fairly different decisions, due to their personal experience, the sailor community (a community is a collection of sailors with similar jobs and trained skills) they handle and the current environmental situation, not to mention emotions and mistakes. Thirdly, the data is highly correlated. For our study we collected data from Navy databases and surveys of Navy datailers. Some attributes in the databases contain overlapping data. Some can be directly calculated from others. Some contain refined information of other attributes. Some are just naturally correlated with each other. Also detailers typically offer no more than one job for sailors, therefore the surveyed decisions correlate too. This makes job assignment a challenging, yet crucial task.

Currently some 280 detailers, handle the job assignment task for some 320,000 enlisted sailors. Although various software programs are routinely used to enhance the process, decisions are made manually. Any technique that can automate the process can prove invaluable for the Navy's personal management unit.

IDA, an Intelligent Distribution Agent, is being developed by the "Conscious" Software Research Group at the University of Memphis. IDA is largely able to automate detailer tasks, including perception and language generation [1]. Yet decision making in IDA is a delicate and ever changing process, much like human decision making.

In order to automate the decision making process in IDA we need a model, which can make optimal or nearly optimal decisions and is able to maintain those over time with little or no human supervision. This requires adaptation to environmental changes (such as economic changes, Navy policy changes, wartime/peacetime changes etc.) including unseen situations. With periodic use of data coming from human detailers, the agent may learn to make better and up to date human-like decisions, which follow environmental changes with some delay. Moreover IDA could also learn to make better decisions through online experience with sailors. However, to test the efficiency of the latter, we need to measure Navy and sailor satisfaction, rather than the percentage of decisions that agree with the detailer provided data.

To obtain good results appropriate data acquisition, preprocessing, and suitable model selection are critical. With certain post-processing we may also further improve model performance. In this paper, various types of neural networks, neuro-fuzzy systems and modern statistical models that learn detailer-like decisions are explored and compared.

In Section 2 we describe how the data was acquired and preprocessed. In Section 3 we discuss several different approaches for decision making: Feedforward Neural Networks, Adaptive Neuro-Fuzzy Inference System, Support Vector Machine and Adaptive Bayes Classifier with Generalized Estimation Equation. Comparison study is provided in section 4 and concluding remarks with future directions in section 5.

2 Data Acquisition and Preprocessing

The data was extracted from the US Navy's Assignment Policy Management System's job and sailor databases. Output data (decisions) were acquired from a Navy detailer in the form of Boolean answers for each match (1 for jobs to be offered, 0 for the rest). For the study one particular community, the Aviation Support Equipment Technicians community was chosen. The databases contained 467 sailors and 167 possible jobs for them. From the more than 100 attributes in each database only those were selected which are most important for the decision making process: eighteen attributes from the sailor database and six from the job database. Various hard constraints (constraints which have to be satisfied) were applied to these attributes complying with Navy policies.

The 1277 matches that passed the hard constraints were inserted into a new database. Every sailor along with all his possible jobs satisfying the hard constraints were assigned to a unique group that is called group ID. This is important because the outputs (decisions given by detailers) were highly correlated: there was typically no more than one job offered to each sailor.

Apart from the hard constraints, there are also soft constraints that need to be satisfied as closely as possible. Soft constraints are designed to increase sailor happiness and to satisfy those Navy policies that are not hard. Each constraint is represented by a function (f_i) with range [0,1], with 1 being the optimal value. The functions were defined and normalized earlier through information given by Navy detailers. The function values served as inputs to the neural networks and other models.

The data was presented to a detailer, who was asked to make decisions (offer jobs) based on the following four widely used soft constraints:

- Job Priority Match: The higher the job priority, the more important it is to fill the job (f_1)
- Sailor Location Preference Match: It is better to send a sailor to a place he/she wants to go (f_2)
- Paygrade match: The sailor's paygrade should exactly match the job's required paygrade (f_3)
- Geographic Location Match: Certain moves are more preferable for the Navy than others (f_4)

Note that decisions were made sequentially and independently. This simulates the typical task a detailer normally faces: offer a job or jobs to a given sailor at the given time from a given list of possible jobs.

3 Methods

3.1 Feedforward Neural Network

3.1.1 Generalized Linear Model (GLM) with logistic link function

One simple case of FFNN topology is one input layer and one output layer with logsigmoid function [5]. This is equivalent to the Generalized Linear Model with logistic function, which can be used to evaluate the relative importance of functions $f_1,...,f_4$ and the conditional probability of the occurrence of the job to be offered.

$$\hat{y} = P(decision = 1 \mid w) = g(w^T f) \qquad (1)$$

where g uses a logistic link function, w is a column vector of weights to be estimated, and f is a column vector of inputs "$f_1,\dots,f_4$".

The coefficients for $f_1,...,f_4$ can be learned using FFNN with the quasi-Newton method [4]. The classification of decisions can be achieved through the best threshold with the largest estimated conditional probability from group data. To setup the best threshold we employed Receiver Operating Characteristic (ROC) to provide the percentage of detections correctly classified and the non-detections incorrectly classified.

3.1.2 Multi Layer Perceptron (MLP)

To improve the generalization performance the Multi Layer Perceptron with one and two hidden layers with structural learning were employed. Network architectures with different degrees of complexity can be obtained through adapting different types of activation functions, number of hidden nodes, and hidden layers. We used the following performance function [2]:

$$J = SSE + \lambda \sum |w| \tag{2}$$

where SSE is the Sum of Squared Error, λ is penalty factor, and $\Sigma|w|$ is complexity penalty term.

To minimize the cost function (2) we used structural learning with forgetting, which can train and prune the network simultaneously, and it yields the best size and structure of the network without the loss of accuracy [11]. By decreasing the effective number of weights during training with forgetting, the danger of overfitting can be reduced. In our study the value of λ in (2) ranged from 0.0001 to 1. Sensitivity analysis was performed through multiple test runs from random starting points to decrease the chance of getting trapped in a local minimum and to find stable results.

For comparison we employed the following learning algorithms: back-propagation with momentum, conjugate gradient, quickprop and delta-delta [6]. The back-propagation with momentum algorithm has the major advantage of speed and is less susceptible to trapping in local minima. Back-propagation adjusts the weights in the steepest descent direction in which the performance function is decreasing most rapidly but it does not necessarily produce the fastest convergence. The search of the conjugate gradient is performed along conjugate directions, which produces generally faster convergence than steepest descent directions. The quickprop algorithm uses information about the second order derivative of the performance surface to accelerate the search. Delta-delta is an adaptive step-size procedure for searching a performance surface.

3.2 Adaptive Neuro-Fuzzy Inference System

Applying a neuro-fuzzy inference system to model this real life problem is quite natural because of the similarity to real life human decision-making. Some design issues were set up based on information from Navy experts, while others were learned or experimented with. For the sake of the comparison study IDA's current real valued satisfaction degrees (f_1,...,f_4 values) for soft constraints were fuzzified. Later we may use linguistic descriptions (low, medium, high, etc.) from human detailers. Even though fuzzification may mean some initial loss of information already included in the f_1,...,f_4 functions, we don't rely on frequent, delicate tuning of those over time as the environment changes.

Based on domain specific knowledge we determined that three triangular membership functions (low, medium, high) for each of the four soft constraints model detailer decisions well, which was verified through experiment. We used a single output, obtained using weighted average defuzzification. All output membership functions had the same type and were either constant or linear (zeroth and first order Sugeno type system). For optimization we used backpropagation and mixed least squares with backpropagation (hybrid) methods. The number of output membership functions ranged from 2 to 243, each of which corresponding to a fuzzy rule. The rules after training were pruned using the rule extraction with Fast Apriori algorithm and combination of rules [3].

Through an adaptive neuro-fuzzy inference system the membership functions are learned, a set of fuzzy rules are created and their weights are adjusted in order to better model the training data. The performance function values are calculated, and classification of decision-making is provided.

3.3 Support Vector Machine

A Support Vector Machine can apply any kind of network structure and typically uses a kernel function with regularization [10]. SVM tries to find a hyperplane in a high dimensional space that separates training samples of each class while maximizing the minimum distance between the hyperplane and any training samples. We employed a Radial Basis Function (RBF) network structure and the Adatron learning algorithm. The advantage of RBF is that it can place each data sample with Gaussian distribution so as to transform the complex decision surface into a simpler surface and then use linear discriminant functions. The Adatron algorithm substitutes the inner product of patterns in the input space with the kernel function of the RBF network. The performance function used was the following:

$$M = \min_{i} J(x_i) \quad (3)$$

$$J(x_i) = \lambda_i \left(\sum_{j=1}^{N} \lambda_j w_j G(x_i - x_j, 2\sigma^2) + b\right) \quad (4)$$

where λ_i is multiplier, w_i is weight, G is Gaussian distribution, σ is standard deviation and b is bias.

We chose a common starting multiplier (0.15), learning rate (0.70), and a small threshold (0.01). While M is greater than the threshold, we choose a pattern x_i to perform the update. After update only some of the weights are different from zero (called the support vectors), they correspond to the samples that are closest to the boundary between classes. The Adatron algorithm uses only those inputs for training that are near the decision surface since they provide the most information about the classification so it can prune and adapt a RBF to have an optimal margin. It provides good generalization and generally yields no overfitting problems, so we do not need the cross-validation set to stop training early.

3.4 Adaptive Bayes Classifier with Generalized Estimation Equation

Adaptive Bayes classifiers use a modified Naive Bayes algorithm [8] and are based on the estimations from GEE with an additive noise model. The test sample of the class can be decided by

$$class(x) = \arg_i \max\{-\log(\sigma_i) - 0.5((x - \mu_i)/\sigma_i)^2\} \quad (5)$$

where μ_i, σ_i are estimated means and standard deviations of each class from GEE with additive model.

GEE models were developed to extend the GLMs to accommodate correlated data [7]. It can be dynamically used to estimate the effects of the changes over time in covariates (1) (values of f1,...,f4) on the response (decision) with the correlated measurement and give us more insight into what criteria are important to decision-makers and the weight of each. Additive noise model added to the GEE was obtained through Gibbs sampling with Bayesian framework [9]. It can deal with noise and outliers (which partially come from the virtually non-deterministic, subjective nature of human decision making) more efficiently and robustly through treating some outputs as missing values and adding prior distribution to the Gibbs sampling. This can overcome drawbacks of the survey and time delay effects on the output 'decision'. Moreover it estimates the noise through assessments of uncertainty in the posterior probabilities of belonging to classes.

The advantage of AB classifiers is that it can identify attributes most useful for classification through GEE model selection according to the Akaike Information Criteria (AIC). It also counts the correlation structure among the outputs into the estimation of the mean and standard deviation of each class. Moreover it drops the assumption of independence made by Naive Bayes algorithms. All these important pieces of information are partially ignored in other classification methods, which may make biased estimation and lower the reliability of the results.

4 Results and Comparison of Methods

For implementation we used a Matlab 6.1 [4] environment with at least a 500 MHz Pentium III processor. For comparison study we repeated some of our experiments with Neurosolutions 4.0, which provided supporting results. For data acquisition and preprocessing we used SQL queries with SAS 8.0.

To evaluate model performance we primarily considered the correct decision making rate for the testing set. In this study a decision was considered to be correct if it was the same as the decision given by the surveyed detailer. For reliable results each experiment was repeated 10 times with 10 different initial weights. The reported values were averaged over the runs.

For neural network training we used up to 5000 epochs, but in most cases there were no significant improvements in the performance function after 1000 epochs. For the reported results in Table I we used 1000 epochs.

GLM with logistic link function gives the weight estimation after normalization for the four coefficients: Job Priority Match, Sailor Location Preference Match, Paygrade Match, Geographic Location Match as follows: 0.316, 0.064, 0.358, 0.262 respectively. These values were used together with IDA's current f_1,...,f_4 functions to provide fitness values.

Through structural learning with forgetting MLP with 15 hidden nodes in one hidden layer worked best. As a learning algorithm delta-delta algorithm provided the best results. As activation function logsig and as the value of λ 0.001 provided the best result. MLP with two hidden layers were also tested but no significant improvement was observed.

For the ANFIS linear output membership functions worked better than constant ones. For optimization function the back-propagation and hybrid method performed similarly regarding the SSE and the classification rate, but the hybrid method's running time was about five times as long as that of the backpropagation's. The ANFIS results in Table I are based on the backpropagation algorithm. Through rule extraction and combination of rules we decreased the number of rules from 81 to 12, which gave nearly as good results as the original rules.

Fig. 1 shows the ANFIS model structure after rule extraction and combination. Its layered structure from left to right is the following: Layer 1. Input neurons: four input neurons for the four soft constraints. Layer 2. Input membership functions: three triangular membership functions for each input neuron. Layer 3. Fuzzy rule left hand sides: each connected to some input membership functions. Layer 4. Output membership functions (right hand sides): the right hand side rules are in one to one relation with the left hand side rules. Layer 5 Aggregated output: each output membership function gets aggregated along with the weight they carry. Layer 6. Output (decision).

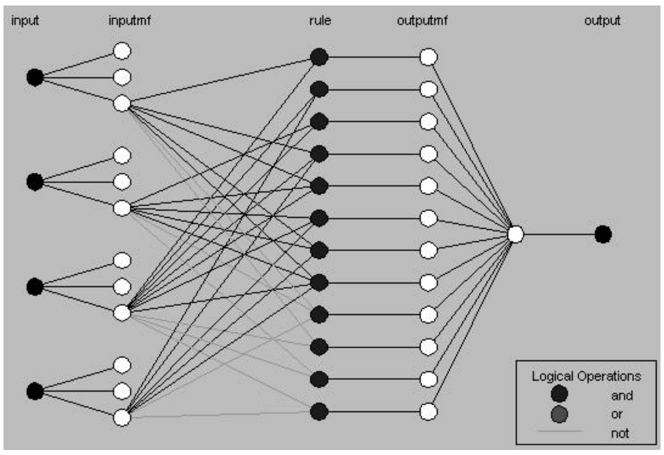

Fig. 1. ANFIS network architecture after rule extraction

SVM with RBF as model and Adatron as learning algorithm performed better than the MLP since the training process focuses on the area around the decision surface. Various versions of RBF networks (spread, error rate, etc.) without SVM were also applied but the results were far less encouraging for generalization than SVM with the Adatron algorithm.

To further improve the classification rates we applied our extra knowledge about the data set. The surveyed detailer typically offered no more than one job to each sailor. Offering only a job with the highest output value could greatly increase the classification rate. Comparison before and after post-processing (using group ID) of the five approaches discussed is shown in Table I. All the reported values are based on 25 % testing set. The table shows that the SVM gave the highest decision making accuracy, but it had relatively high time cost. The performances of AB with GEE and MLP with 15 nodes are also encouraging. MLP also had a very low time cost. The relatively high decision making capability of the ANFIS gives us a reasonable way for future sampling of detailers, which doesn't depend on the delicate noisy tuning of the functions for soft constraints, but its high time cost may make it a poor choice for large data sets. Although the decision-making capability of the GLM with logistic link function is the lowest among the reported methods, its low time and architecture complexity may make it a good model candidate in case a large data set becomes available in the future. Moreover this is the method, which mostly exploits the exact definitions of the $f_1,...,f_4$ functions, so if we can somehow improve the definition of these functions, this model may become a very useful tool for decision making purposes.

Some combination of the provided methods may also be useful to make up the weaknesses of one another. For example, the structural learning with forgetting in MLP generally has good approximation, robustness to noise and requires no preliminary knowledge. However, it provides no explanation of the obtained results, which could be done with fuzzy rules generated by the ANFIS.

Table 1. average Running Times (RT) in seconds, average Correct Decision Making Rates (CDMR) and average Correct Decision Making Rates after Post Processing (PP) for different methods (average of 10 runs)

Method	RT	CDMR	PP
GLM	**170**	87.4%±0.1%	91.2%±0.2%
MLP	227	92.3%±0.1%	94.3%±0.2%
ANFIS	3050	91.5%±0.4%	93.2%±0.3%
SVM	1425	**94.5%±1.0%**	**95.5%±0.6%**
AB	1035	93.4%±0.2%	95.2%±0.4%

Naturally, we can't expect 100% correct classification rate because of the presence of noise from various sources. (Note that in about 1% of the cases the surveyed detailer offered 2 or more jobs to sailors, which couldn't be overcome with our postprocessing.) Most important perhaps is the indeterminate nature of detailer decisions. Even the same detailer may make different decisions on the same data at different times. It is widely believed that different detailers are also likely to make different decisions even under the same circumstances. Moreover environmental changes further bias decisions, so periodic training on up to date data sets is necessary in order to keep decision making up to date in IDA. As a result our reported and observed values can't tell us how well our system makes decisions, but only how well it follows the decisions given by the surveyed detailers. This problem is present in Navy practices as well: they don't have a well-defined formula to effectively evaluate detailer performance.

5 Conclusion

In this paper we presented various neural network and statistical learning approaches in order to approximate the human decision-making process for job assignment of the US Navy. Results show that the Support Vector Machine is capable of making the highest quality of human like decisions, although with high standard deviation and time cost. Other methods also performed well, and they can be reasonable choices for future needs in IDA, each offering some advantage, such as running time, robustness, and model complexity, over the others. With the use of the knowledge of internal correlation of the data, postprocessing further improved performance for all models. However, no matter which model we choose to place online as part of a possible IDA product, data needs to be provided to the agent periodically in order to allow up to date decision-making through learning. Some learning can also take place based on the agent's own experience of the sailor behavior. Learning can also take place based on command behavior and feedback from various sources well after assignments have been made. Test of this learning could provide information if the agent is able to act as a human detailer in the decision making environment. Success of such a test not only depends on the performance of the decision-making module, but on all the integrated modules of the agent.

Acknowledgement

This work is supported in part by ONR grant (no. N00014-98-1-0332). The authors thank E. Olusegun George, Robert Kozma, and King-Ip Lin for their contribution to this paper, as well as the members of the "Conscious" Software Research Group who developed IDA.

References:

[1] S. Franklin, A. Kelemen and L. McCauley, IDA: A cognitive agent architecture, *In the proceedings of IEEE International Conference on Systems, Man, and Cybernetics '98*, IEEE Press, 1998, pp. 2646.

[2] R. Kozma, M. Sakuma, Y. Yokoyama and M. Kitamura, On the Accuracy of Mapping Backpropagation with Forgetting, *Neurocomputing*, Vol. 13, No. 2-4, 1996, pp. 295-311.

[3] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen and A. Verkamo, Fast Discovery of Associstion Rules, *in Advances in Knowledge Discovery and Data Mining, MIT Press,* 1996, pp. 307-328.

[4] *Matlab User Manual,* Release 6.0, Natick, MA: MathWorks, Inc., 2001.

[5] M. Schumacher, R. Rossner and W. Vach, Neural networks and logistic regression: Part I', *Computational Statistics and Data Analysis,* 21, 1996, pp. 661-682.

[6] L. H. Tsoukalas and R. E. Uhirg, *Fuzzy and Neural Approaches in Engineering,* John Wiley & Sons, N. Y., 1997.

[7] K. Y. Liang and S. L. Zeger, Longitudinal Data Analysis Using Generalized Linear Models, *Biometrika,* 73, 1986, pp. 13-22.

[8] A. D. Keller, M. Schummer, L. Hood and W. L. Ruzzo, Bayesian Classification of DNA Array Expression Data, *Technical Report UW-CSE-2000-08-01,* 2000.

[9] A. E. Gelfand, S. E. Hills, A. Racine-Poon and A. F. M. Smith, Illustration of Bayesian inference in normal data models using Gibbs sampling, *Journal of the American Statistical Association,* 85 (412), 1990, pp. 972–985.

[10] T. T. Friess, N. Cristianini and C. Campbell, The kernel adatron algorithm: a fast and simple learning procedure for support vector machine, *In Proc. 15th International Conference on Machine Learning, Morgan Kaufman Publishers,* 1998.

[11] D. A. Miller and J. M Zurada, A dynamical system perspective of structural learning with forgetting, *IEEE Transactions on Neural Networks,* vol. 9, no. 3, 1998, pp. 508-515.