Proceedings of the 5th WSEAS Int. Conf. on SIMULATION, MODELING AND OPTIMIZATION, Corfu, Greece, August 17-19, 2005 (pp256-261)

Optimizing Decision Making with Neural Networks in Software Agents

ARPAD KELEMEN[1,2,3]
YULAN LIANG[1,2]
[1] Department of Mathematical Sciences, University of Memphis
[2] Department of Biostatistics, State University of New York at Buffalo
[3] Department of Computer and Information Sciences, Niagara University
249 Farber Hall, 3435 Main Street, Buffalo, NY 14214
USA
purple.niagara.edu/akelemen

Abstract: Finding suitable jobs for US Navy sailors from time to time is an important and ever-changing process. An Intelligent Distribution Agent (IDA) and particularly its constraint satisfaction module take up the challenge to automate the process. The constraint satisfaction module's main task is to provide the bulk of the decision making process in assigning sailors to new jobs in order to maximize Navy and sailor "happiness". We propose Multilayer Perceptron neural network with structural learning in combination with several statistical criteria to aid IDA's constraint satisfaction module, which is also capable of learning high quality decision making over time. Data were taken from Navy databases and from surveys of Navy experts. Results show highly accurate classification and encouraging prediction.

Key-Words: Decision making, Optimization, Multilayer perceptron, Structural learning, Software agent

1 Introduction

IDA [1], is a "conscious" [2], [3] software agent [4], [5] that was built for the U.S. Navy by the Conscious Software Research Group at the University of Memphis. IDA was designed to play the role of Navy employees, called detailers, who assign sailors to new jobs from time to time. One of its modules, the constraint satisfaction module, was responsible for satisfying constraints to ensure the adherence to Navy policies and sailor preferences. The model employed a linear functional to assign fitness values to each candidate job for each candidate sailor. The functional yielded a value in [0,1] with higher values representing higher degree of "match" between the sailor and the job. Some of the constraints were soft, while others were hard. Soft constraints can be violated without invalidating the job. Associated with the soft constraints were functions which measured how well the constraints were satisfied for the sailor and the given job at the given time, and coefficients which measured how important the given constraint was relative to the others. The hard constraints cannot be violated.

The process of using this method for decision making involves periodic tuning of the coefficients and the functions. A number of alternatives and modifications have been proposed, implemented and tested for large size real Navy domains [6-8]. These techniques are optimization tools that yield an optimal solution or one which is nearly optimal. Most of these implementations were performed, by other researchers, years before the IDA project took shape and according to the Navy they often provided low rate of "matching" between sailors and jobs. This showed that standard operation research techniques are not easily applicable to this real life problem if we are to preserve the format of the available data and the way detailers currently make decisions. High quality decision making is an important goal of the Navy but they need a working model that is capable of making decisions similarly to a human detailer under time pressure, uncertainty, and is able to learn/evolve over time as new situations arise and new standards are created. For such task, clearly, an intelligent agent (IDA) and a learning neural network is better suited than standard operation research tools. At this point we want to tune the functions and their coefficients in the constraint satisfaction module as opposed to trying to find an optimal solution for the decision making problem in general. Finally, detailers, as well as IDA, receive one problem at a time, and they try to find a job for one sailor at a time. Simultaneous job search for multiple sailors is not a current goal of the Navy or IDA. Instead, detailers (and IDA) try to find the “best” job for the “current” sailor all over the time. Our goal in this paper is to use neural networks and statistical methods to learn from Navy detailers, and to enhance decisions made by IDA's constraint satisfaction module. The functions for the soft constraints were set up semi-heuristically in consultation with Navy experts. We will assume that they are optimal, though future efforts will be made to verify this assumption.

While human detailers can make judgments about job preferences for sailors, they are not always able to quantify such judgments through functions and coefficients. Using data collected periodically from human detailers, a neural network learns to make human-like decisions for job assignments. It is important to set up the functions and the coefficients in IDA to reflect the characteristics of the human decision making process. A neural network gives us more insight on what preferences are important to a detailer and how much. Moreover inevitable changes in the environment will result changes in the detailer's decisions, which could be learned with a neural network although with some delay.

In this paper, we propose MLP with structural learning for achieving optimal decisions in software agents. The job assignment problem of other military branches may show certain similarities to that of the Navy, but the Navy's mandatory “Sea/Shore Rotation” policy makes it unique and perhaps, more challenging than other typical military, civilian, or industry types of job assignment problems. Unlike in most job assignments, the Navy sends its sailors to short term sea and shore duties periodically, making the problem more constrained, time demanding, and challenging. This was one of the reasons why we designed and implemented a complex, computationally expensive, human-like “conscious” software. This software is completely US Navy specific, but it can be easily modified to handle any other type of job assignment.

In Section 2 we describe how the data were attained and formulated into the input of the neural network. In Section 3 we discuss FFNNs with Logistic Regression, performance function and statistical criteria of MLP Selection for best performance including learning algorithm selection. Section 4 presents some comparative analysis and numerical results.

2 Data Acquisition

The data was extracted from the Navy's Assignment Policy Management System's job and sailor databases. For the study one particular community, the Aviation Support Equipment Technicians (AS) community was chosen. The databases contained 467 sailors and 167 possible jobs for the given community. From the more than 100 attributes in each database only those were selected which are important from the viewpoint of the constraint satisfaction: Eighteen attributes from the sailor database and six from the job database. For this study we chose 4 soft constraints. 1277 matches passed the mandatory hard constraints, which were inserted into a new database.

Table 1 shows the four soft constraints applied to the matches that satisfied the hard constraints and the functions which implement them. These functions measure degrees of satisfaction of matches between sailors and jobs, each subject to one soft constraint.

Table 1.

	Policy name	Policy
f_1	Job Priority	High priority jobs are more important to be filled
f_2	Sailor Location Preference	It's better to send a sailor where he/she wants to go
f_3	Paygrade	Sailor's paygrade should match the job's paygrade
f_4	Geographic Location	Certain moves are more preferable than others

Output data (decisions) were acquired from an actual detailer in the form of Boolean answers for each possible match.

3 Design of Neural Network

One natural way the decision making problem in IDA can be addressed is via the tuning of the coefficients for the soft constraints. This will largely simplify the agent's architecture, and it saves on both running time and memory. Decision making can also be viewed as a classification problem, for which neural networks demonstrated to be a very suitable tool. Neural networks can learn to make human-like decisions, and would naturally follow any changes in the data set as the environment changes, eliminating the task of re-tuning the coefficients.

3.1 Feedforward Neural Network with Logistic Regression

We use a logistic regression model to tune the coefficients for the functions $f_1,...,f_4$ for the soft constraints and evaluate their relative importance. The corresponding conditional probability of the occurrence of the job to be offered is

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

$$g(a) = \frac{e^a}{1+e^a} \qquad (2)$$

where g represents the logistic function evaluated at activation a. Let w denote weight vector and f the column vector of the importance functions: $f^T = [f_1,..., f_4]$. Then the "decision" is generated according to the logistic regression model.

The weight vector w can be adapted using FFNN topology [9], [10]. In the simplest case there is one input layer and one output logistic layer. This is equivalent to the generalized linear regression model with logistic function. The estimated weights satisfy Eq.(3).

$$\sum_i w_i = 1,\ 0 \le w_i \le 1 \qquad (3)$$

The linear combination of weights with inputs $f_1,...,f_4$ is a monotone function of conditional probability, as shown in Eq.(1) and Eq.(2), so the conditional probability of job to be offered can be monitored through the changing of the combination of weights with inputs $f_1,...,f_4$. The classification of decision can be achieved through the best threshold with the largest estimated conditional probability. The class prediction of an observation x from group y was determined by

$$C(x) = \arg\max_k \Pr(x \mid y = k) \tag{4}$$

To find the best threshold we used Receiver Operating Characteristic (ROC) to provide the percentage of detections correctly classified and the non-detections incorrectly classified. To do so we employed different thresholds with range in [0,1]. To improve the generalization performance and achieve the best classification, the MLP with structural learning was employed [11], [12].

3.2 Neural Network Selection

Since the data coming from human decisions inevitably include vague and noisy components, efficient regularization techniques are necessary to improve the generalization performance of the FFNN. This involves network complexity adjustment and performance function modification. Network architectures with different degrees of complexity can be obtained through adapting the number of hidden nodes and partitioning the data into different sizes of training, cross-validation and testing sets and using different types of activation functions. A performance function commonly used in regularization, instead of the sum of squared error (SSE) on the training set, is a loss function (mostly SSE) plus a penalty term [13]-[16]:

$$J = SSE + \lambda \sum w^2 \tag{5}$$

From another point of view, for achieving the optimal neural network structure for noisy data, structural learning has better generalization properties and usually use the following modified performance function [11], [12]:

$$J = SSE + \lambda \sum |w| \tag{6}$$

Yet we propose an alternative cost function, which includes a penalty term as follows:

$$J = SSE + \lambda n / N \tag{7}$$

λ is a penalty factor, n is the number of parameters in the network and N is the size of the input. This helps to minimize the number of parameters (optimize network structure) and improve the generalization performance.

In our study the value of λ in Eq.(7) ranged from 0.01 to 1.0. Note that λ=0 represents a case where we don't consider structural learning, and the cost function reduces into the sum of squared error. Normally the size of input samples should be chosen as large as possible in order to keep the residual as small as possible. Due to the cost of the large size samples, the input may not be chosen as large as desired. However, if the sample size is fixed then the penalty factor combined with the number of hidden nodes should be adjusted to minimize Eq.(7).

Since n and N are discrete, they can not be optimized by taking partial derivatives of the Lagrange multiplier equation. For achieving the balance between data-fitting and model complexity from the proposed performance function in Eq.(7), we would also like to find the effective size of training samples included in the network and also the best number of hidden nodes for the one hidden layer case. Several statistical criteria were carried out for this model selection in order to find the best FFNN and for better generalization performance. We designed a two-factorial array to dynamically retrieve the best partition of the data into training, cross-validation and testing sets with adapting the number of hidden nodes given the value of λ:

- Mean Squared Error (*MSE*)
- Correlation Coefficient (r)
- Akaike Information Criteria (AIC)
- Minimum Description Length (MDL)

The *MSE* can be used to determine how well the predicted output fits the desired output. More epochs generally provide higher correlation coefficient and smaller MSE for training in our study. To avoid overfitting and to improve generalization performance, training was stopped when the MSE of the cross-validation set started to increase significantly. Sensitivity analyses were 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.

The network with the lowest *AIC* [17] or *MDL* [18] is considered to be the preferred network structure. The choice of the best network structure is based on the maximization of predictive capability, which is defined as the correct classification rate and the lowest cost given in Eq.(7).

4 Data Analysis and Results

For implementation we used a Matlab 6.1 [19] environment with at least a 1GHz Pentium IV processor. For data acquisition and preprocessing we used SQL queries with SAS 9.0.

4.1 Estimation of Coefficients

FFNN with back-propagation with momentum with logistic regression gives the weight estimation for the four coefficients as follows: 0.316, 0.064, 0.358, 0.262 respectively. Simultaneously, we got the conditional probability for decisions of each observation from Eq.(1). We chose the largest estimated logistic probability from each group as predicted value for decisions equal to 1 (job to be offered) if it was over threshold. The threshold was chosen to maximize performance and its value was 0.65. The corresponding correct classification rate was 91.22% for the testing set. This indicates a good performance.

4.2 Neural Network for Decision Making

Multilayer Perceptron with one hidden layer was tested using tansig and logsig activation functions for hidden and output layers respectively. Other activation functions were also used but did not perform as well. MLP with two hidden layers were also tested but no significant improvement was observed. Four different learning algorithms were applied for sensitivity analysis. For reliable results and to better approximate the generalization performance for prediction, each experiment was repeated 10 times with 10 different initial weights. Training was confined to 5000 epochs, but in most cases there were no significant improvement in the *MSE* after 1000 epochs. The best MLP was obtained through structural learning where the number of hidden nodes ranged from 2 to 20, while the training set size was setup as 50%, 60%, 70%, 80% and 90% of the sample set. The cross-validation and testing sets each took the half of the rest.

5 Conclusion

High-quality decision making using optimum constraint satisfaction is an important goal of IDA, to aid the Navy to achieve the best possible sailor and Navy satisfaction performance. A number of neural networks with statistical criteria were applied to either improve the performance of the current way IDA handles constraint satisfaction or to come up with alternatives. IDA's constraint satisfaction module, neural networks and traditional statistical methods are complementary with one another. In this work we proposed and combined MLP with structural learning, statistical criteria, and a novel cost function, which provided us with the best MLP with one hidden layer. Coefficients for the existing IDA constraint satisfaction module were adapted via FFNN with logistic regression. It is important to keep in mind that the coefficients have to be updated from time to time as well as online neural network trainings are necessary to comply with changing Navy policies and other environmental challenges.

References:

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

[2] B. J. Baars, *Cognitive Theory of Consciousness*, Cambridge University Press, Cambridge, 1988.

[3] B. J. Baars, *In the Theater of Consciousness*, Oxford University Press, Oxford, 1997.

[4] S. Franklin and A. Graesser, Intelligent Agents III: Is it an Agent or just a program?: A Taxonomy for Autonomous Agents, *Proceedings of the Third International Workshop on Agent Theories, Architectures, and Languages*, Springer-Verlag, pp. 21-35, 1997.

[5] S. Franklin, *Artificial Minds*, Cambridge, Mass, MIT Press, 1995.

[6] R. Kondadadi, D. Dasgupta, and S. Franklin, An Evolutionary Approach For Job Assignment,*Proceedings of International Conference on Intelligent Systems*, Louisville, Kentucky, 2000.

[7] T. T. Liang and T. J. Thompson, Applications and Implementation - A large-scale personnel assignment model for the Navy, *The Journal For The Decisions Sciences Institute*, Volume 18, No. 2 Spring, 1987.

[8] T. T. Liang and B. B. Buclatin, Improving the utilization of training resources through optimal personnel assignment in the U.S. Navy, *European Journal of Operational Research* 33, pp. 183-190, North-Holland, 1988.

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

[10] E. Biganzoli, P. Boracchi, L. Mariani, and E. Marubini, Feed Forward Neural Networks for the Analysis of Censored Survival Data: A Partial Logistic Regression Approach, *Statistics in Medicine*, 17, pp. 1169-1186, 1998.

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

[12] M. Ishikawa, Structural learning with forgetting, *Neural Networks*, Vol. 9, pp. 509-521, 1996.

[13] S. Haykin, *Neural Networks*, Prentice Hall Upper Saddle River, New Jersey, 1999.

[14] F. Girosi, M. Jones, and T. Poggio, Regularization theory and neural networks architectures, *Neural Computation*, 7:219--269, 1995.

[15] Y. Le Cun, J. S. Denker, and S. A. Solla, Optimal brain damage, in D. S. Toureczky, ed. *Adavnces in Neural Information Processing Systems 2* (Morgan Kaufmann), pp. 598-606, 1990.

[16] C. M. Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press, 1995.

[17] H. Akaike, A new look at the statistical model identification, *IEEE Trans. Automatic Control*, Vol. 19, No. 6, pp. 716-723, 1974.

[18] J. Rissanen, Modeling by shortest data description, *Automat.*, Vol. 14, pp. 465-471, 1978.

[19] Matlab2004 *Matlab User Manual*, Release 6.0, Natick, MA: MathWorks, Inc, 2004.