Developing a Model Selection Method for Real Time Decision Support Systems

Ching-Chang Lee
Department of Information Management, Shu-Te University, cclee@mail.stu.edu.tw

Shing-Hwang Doong
Department of Information Management, Shu-Te University, tungsh@mail.stu.edu.tw

Abstract

This paper develops a model selection method for real time decision support system (RTDSS). In this paper, we consider the parameter precision as a function of time *t* and cost *c* and then our model selection criterion will be the one to choose the model from available ones for solving a problem that offers the highest payoff value under the given time and cost constraints. Since the payoff function is now a nonlinear function of several variables under various constraints, the (*t, c*) solution space is complex and a genetic algorithm is employed to find the maximum value of the payoff function. Based on the results, we also propose a model selection system that is flexible to implement with the help of web services technology.

Keywords: Decision support system, Model management system, Real time decision support system (RTDSS), Model selection, Genetic algorithms, Web services

1. INTRODUCTION

The decision paradigm has changed from periodical decision-making into real-time decision-making to quickly respond to changing environments (Trebilcoke 2000). Roy (2001) believes that as enterprises could obtain necessary information quickly with the help of the integration of Internet and enterprise information system, and that if an appropriate decision model is provided, enterprises could make computer based real-time decisions. Therefore, how to improve the existing Decision Support Systems (DSS) or even develop Real-time Decision Support Systems (RTDSS) to solve real-time problems has become an important issue in DSS disciplines.

Dealing with a real-time problem, a decision-maker should select a fittest model and use the model to solve his problem within time constraints. Selecting of a model from alternative models becomes critical to decision makers, especially when the decision must be made within the time constraints. Most researches in Model Management System (MMS) focused on supporting model creation, storage, retrieval, execution, maintenance, representation, and model base organization (Bonczek et al. 1981; Liang and Jones 1988). In addition to these issues, researchers believed that it is very important for an MMS to have the capability of model selection (Liang 1988; Mayer 1998). The purpose of the capability of model selection is to help the user determine what models are available to produce the requested information and then automatically select or allow the user to select a model for execution (Liang 1988).

A model can accept as inputs the data extracted from the database subsystem of a DSS or the information collected from the outside world such as the Internet. There is no doubt that the precision of input data will affect the solution quality of a model solver. There is also no doubt that the precision of a model parameter will affect the output solution of a model. Since most models are established based on empirical data, noises can affect the precision of a model parameter, which will affect the final solution quality ultimately.

In this paper, we consider the parameter precision as a function of time t and cost c. Our model selection criterion will be the one to choose the model from available ones for solving a problem that offers the highest payoff value under the given time and cost constraints. Since the payoff function is now a nonlinear function of several variables under various constraints, the (t, c) solution space is complex and a genetic algorithm is employed to find the maximum value of the payoff function.

This paper is organized as follows: In section 2, we describe the problem of model selection by defining some terms and concepts used in this study. The payoff function and many key ingredients for understanding the payoff function are explained in this section. A scenario study of the model selection problem is given in section 3. Based on the scenario study, we propose a model selection system that is flexible to implement with the help of web services technology in section. Conclusions and future research directions for model selection

2. MODEL SELECTION PROBLEM

Many factors affect the accuracy of a model solution. However, we make some assumptions in this section before proceeding to the model selection problem. First of all, we do not consider the effect of incorrect input data and we assume that a model solver can provide a solution faithfully for a given set of input data. In other words, the precision of a model solution is assumed to depend on the model parameters only. Secondly, when considering the model cost, we use the cost for obtaining the model parameters only. That is, the added cost for running the model solver is not included here. We introduce additional concepts and terms in the following.

2.1 Model parameter precision function

According to our assumption, the model parameter precision plays an important role in determining the model solution accuracy. A model may have many parameters ($X_1, \ldots, X_n$) that describe the characteristics of the model. The precision of a parameter is defined to be 1 minus the relative deviation of the obtained value from the true value, i.e. precision of parameter (Xi) is equal to

$$Y_i \;=\; 1 - \;| X_i' \; - X_i0 \,| \,/\, | X_i0 \,| \tag{1}$$

where X_i' = obtained value of X_i and X_i0 = true value of X_i. It is reasonable to see that different efforts spent for obtaining a parameter will yield different levels of parameter precision. While there is no standard way to measure the efforts, time and cost are the two most important factors used in business decision. Thus we assume that the precision Y_i of a parameter X_i is a function of time t and cost c_i spent for obtaining the parameter.

$$Y_i = f_i(t, c_i) \tag{2}$$

This parameter precision function has a few characteristics, e.g. it is between 0 and 1, and a higher value of t or c_i will make a higher value of Y_i, i.e. it is a non-decreasing function of the time and cost variables. Moreover, one can make trade off between the time and cost variables for obtaining the desired precision level. In other words, if an enterprise is willing to spend more money, it can obtain the information with the same level of precision for less time. A typical example of the precision function satisfying the above characteristics is given in the following:

$$f_i(t, c_i) = \alpha_i t^{\gamma_i} c_i + \beta_i \tag{3}$$

where $\alpha_i > 0$ is used to adjust the weight between time t and cost c_i, $\beta_i \geq 0$ is the default precision level (without spending any money or time), and $\gamma_i > 0$ controls how fast the cost decays with the increasing time spent for the parameter value. The variables t and c_i are nonnegative, and we are primarily interested in those combinations of the variables to make this precision function between 0 and 1. When certain combinations of t and c_i make the function value greater than one, we will assume the precision value has achieved the maximum level of precision of 1. The coefficients $\alpha_i, \beta_i, \gamma_i$ may be obtained via a suitable regression method with the empirical data. Of course, other forms of precision functions may be assumed if they have the characteristics stated above; and finally genetic programming may be used to do the symbolic regression for finding the parameter precision function from the empirical data (Koza et al. 1999).

2.2 Model solution precision function

Since we assume that a model solver can provide a solution faithfully for any input data and we do not consider the effect of incorrect input data, the precision of a model solution is a function of the model parameters precision:

$$P = g(Y_1, \ldots Y_n) \tag{4}$$

where Y_i is the precision level of the i-th parameter of the model. Again, a couple of assumptions are made for this precision function. First, it is between 0 and 1 with 1 for a perfect model solution, and 0 for a totally imprecise model solution. Secondly, it is a non-decreasing function of the variables, i.e. when we have a higher precision level of the model parameters, we have a higher precision level of the model solution. For example, the following formulae satisfy these conditions for models with 2 or 3 parameters:

$$P = (Y_1^2 + Y_2^2)/2 \tag{5}$$

$$P = Y_1 Y_2 Y_3 \tag{6}$$

Besides satisfying the conditions stated above, these formulae have the following additional characteristics: (i) when all model parameters have a precision level of 0, the model solution has a precision level of 0 too; and (ii) when all model parameters have the highest precision level of 1, the model solution also has the highest level of precision level of 1. Of course, other formulae can be used for the model precision functions, and many times the formulae may be obtained by symbolic regression using genetic programming (Koza et al. 1999).

2.3 Model accuracy and model solution accuracy

Model solution precision describes the precision level of a particular solution offered by the model solver with a specific level of parameter precision. However, how well the model fits the real problem is another issue. For example, a linear programming model may be used to describe a decision problem and high quality of model parameters may be obtained for this model. Then, a high precision level of the model solution can be expected. Now suppose the decision problem is actually a quadratic problem, then the selected linear model will not fit the real problem very well and it receives a low level of model accuracy.

The model accuracy number (Am) is defined to be a quantity between 0 and 1 that explains how well a model fits a real problem. There are possibly many ways to define this quantity. By balancing the ease of computation and the trait of model accuracy, we adopt the following heuristic formula in Lee (2002) for model accuracy in this study:

$$A_m = \left(\sum_{i=1}^{10} A_{-i}\right)/10 \tag{7}$$

where A_{-i} denotes the model accuracy evaluated by the decision makers in the prior *i*-th application of the model to a similar problem. Now we can define the model solution accuracy as follows:

$$A_s = A_m g(Y_1, \ldots, Y_n) \tag{8}$$

Thus, the accuracy of a particular model solution depends both on its solution precision g and the internal model accuracy Am.

2.4 Sensitivity and payoff function

Sensitivity analysis can help decision-makers understand how close a model is related to the real world, and the interaction among variables of a model (Mészáros and Rapcsák 1996; Ringuest 1997). A model solution may have a high level of precision at a particular precision level of model parameters $Y_1^{(0)}, \ldots, Y_n^{(0)}$. In reality, we often need to consider the perturbation effect of the model parameters as well. That is, when the obtained parameters are perturbed a little bit so that the parameter precisions vary slightly, how well the model solution precision is preserved. In order to answer the question, we need to estimate the following term:

$$\Delta g = g(Y_1^{(0)} + \varepsilon_1, \ldots, Y_n^{(0)} + \varepsilon_n) - g(Y_1^{(0)}, \ldots, Y_n^{(0)}) \tag{9}$$

According to the Taylor series expansion, we have

$$|\Delta g| \approx |\nabla g(Y_1^{(0)}, \ldots, Y_n^{(0)})| \sqrt{\varepsilon_1^2 + \cdots + \varepsilon_n^2} \tag{10}$$

where $\nabla g = (\partial g / \partial Y_1, \ldots, \partial g / \partial Y_n)$ denotes the gradient vector of g and the square root expression in equation (10) indicates the strength of the perturbation. Hence, it is natural to define the sensitivity of the solution precision at a particular level of parameter precision as the length of the gradient vector of the solution precision function at that level of parameter precision. That is,

$$S(Y_1^{(0)}, \ldots, Y_n^{(0)}) = |\nabla g(Y_1^{(0)}, \ldots, Y_n^{(0)})| \tag{11}$$

Notice that when this value is 0, the model solution precision is quite resistant to a small perturbation of the obtained parameter values. On the other hand, when the value is high, then a small perturbation in the model parameter values may cause a large variation in the model solution precision.

Having defined the sensitivity of the solution precision, we now define the payoff function of a model as the difference between the revenue and the cost generated by taking an action provided by the model solution in the following:

$$PO = Be^{hA_s}e^{-kS} - Cost \tag{12}$$

where B, h and k are model dependent constants, A_s the model solution accuracy, S the solution precision sensitivity and *Cost* the total information cost for obtaining the model parameters. Equation (12) says that if a higher model solution accuracy A_s can be obtained, then a higher payoff value will be obtained. On the other hand, a higher sensitivity S may reduce the payoff value faster than a lower sensitivity does. Utilizing equations (2), (4), (8) and (11), we can rewrite (12) as follows:

$$PO(t, c_1, \ldots, c_n) = Be^{hA_m g(t,c_1,\ldots,c_n) - k|\nabla g|(t,c_1,\ldots,c_n)} - \sum_{i=1}^{n} c_i \tag{13}$$

With the above assumption for models, we define the objective value of a model to be the maximum payoff value in equation (13) under the following constraints:

$$0 \le t \le T_{\max} \tag{14}$$

$$\sum_{i=1}^{n} c_i \le C_{\max}; c_i \ge 0, i = 1, \ldots n \tag{15}$$

Equation (14) indicates that the time interval for obtaining the model parameters is finite, and equation (15) shows that the total cost for obtaining these parameters is finite too. When alternative models are available for a problem, we assume that the time constraint $T_{\max}$ and the cost constraint $C_{\max}$ are the same for all these competing models, and the model with the highest objective value will be selected for solving the problem. Though one may trade in time with cost for the same level of parameter precision (cf. Figure 1), we assume that the precision value is always between 0 and 1, i,e, $0 \le f_i(t, c_i) \le 1, i = 1\ldots, n$. At times, if a combination of t and c_i satisfying constraints (14) and (15) makes $f_i(t, c_i)$ greater than one, we will assume that the parameter precision value is equal to one for this combination of variables.

Equations (13-15) constitute a complex nonlinear optimization problem with a solution space in $(t, c_1, \ldots, c_n)$. When we have a few alternative models to choose from, we need to solve such an optimization problem for each of these models. Each model may have its own *B, h* and k constants, and f_i and g functions. Thus, an efficient method for solving this type of optimization problem is needed for us to select the best model in the shortest possible time.

Since the payoff function is now a nonlinear function of several variables under various constraints, the (t, c) solution space is complex. A genetic algorithm can be employed to find the maximum value of the payoff function.

3. A SCENARIO STUDY OF MODEL SELECTION PROBLEM

In this section, we describe a scenario for selecting the best model from competing ones for solving a decision problem. A model will be judged by its objective value defined in section 2.4. The model with the highest objective value is selected and recommended to the decision maker. Evaluating the objective value of a model requires the solution of an optimization problem in equations (13) – (15). The genetic algorithm will be used for this optimization purpose. We describe the problem setting in section 3.1 and present the result and discussion in section 3.2.

3.1 The scenario description

To illustrate our model selection procedure, we assume that there are two alternative models that can be used in a decision making process. Model 1 has two parameters while model 2 has three. All related constants and (parameter and model solution) precision functions have been determined either by symbolic regression or heuristics from empirical data. These data are summarized in Table 1. We will consider $T_{\max}$ and $C_{\max}$ in two different cases (Tables 2 and 3).

Table 1 Parameter setting of the example scenario

Model 1	Model 2
A_m=0.6, B = 8000, h = 5, k = 0.5 $Y_1 = f_1(t,c_1) = 0.001tc_1 + 0.5$ $Y_2 = f_2(t,c_2) = 0.0001t^2c_2 + 0.5$ $g(Y_1,Y_2) = (Y_1^2 + Y_2^2)/2$	A_m=0.8, B = 10000, h = 5, k = 0.5 $Y_1 = f_1(t,c_1) = 0.0001tc_1 + 0.5$ $Y_2 = f_2(t,c_2) = 0.0001tc_2 + 0.5$ $Y_3 = f_3(t,c_3) = 0.0001tc_3 + 0.5$ $g(Y_1,Y_2,Y_3) = Y_1Y_2Y_3$

3.2 The result and discussion for model selection

Depending on the business requirement, we may design appropriate chromosomes in a GA using different types of data for genes. For example, if we only consider integer values for time and cost, then a chromosome with all integer genes may be used in the GA. In this study, we assume that both time and cost variables can take non-integer values, so we will use real coded genes in a chromosome. For example, in model one a chromosome is an array of the type (t, c_1, c_2) where t, c_1, c_2 are nonnegative real values satisfying the constraints in equations (14-15). The fitness value of a chromosome is provided by equation (13), which will guide the evolutionary search path for the GA. We use a parallel GA library PGAPACK from the Argonne Nat. Lab. (www-fp.mcs.anl.gov/CCST/research/reports_pre1998/comp_bio/stalk/pgapack.html) to implement the GA. An initial population of 20 chromosomes is generated randomly. A 5000-generation criterion is used to stop the iteration of the GA. Default settings regarding the selection, crossover and mutation operators in PGAPACK are adopted in the calculation.

Since GA is a stochastic search method, several runs of the same program with different initial population are normally performed and the best result from these runs is reported as a near optimal solution of the problem. We ran the GA program 10 times each for these two models, and the best result is reported in Tables 2 and 3. From these tables, we see that model 1 is selected when $T_{\max}$=1 and $C_{\max}$=10000 (Table2), and model 2 is selected when $T_{\max}$=0.5 and $C_{\max}$=100000 (Table 3).

Table 2 Best result from 10 runs of GA when $T_{\max}$=1, $C_{\max}$=10000

	Model 1	Model 2
Objective value	**73720**	45389
Optimizing parameters	t=1, c_1=500, c_2=5007	t=1, c_1=3275, c_2=3437, c_3=3286

Table 3 Best result from 10 runs of GA when $T_{\max}$=0.5, $C_{\max}$=100000

	Model 1	Model 2
Objective value	58215	**199574**
Optimizing parameters	t=0.5, c_1=1000, c_2=20011	t=0.5, c_1=10035, c_2=10014, c_3=10027

Even though model 2 has a higher model accuracy value (0.8 vs. 0.6) than model 1, the actual model selection depends on the time and cost constraints as well. For example, with a tighter budget and longer time available for obtaining the model parameters, model 1 will be selected because it has a higher objective value (73720 vs. 45389) than model 2 (cf. Table 2). On the other hand, when the company has a higher budget and shorter time available for obtaining the model parameters, then model 2 is selected since it has a higher objective value (199574 vs. 58215) than model 1 (cf. Table 3). These tables not only show the comparison results between models, but also the differences caused by different constraints. For example, based on the objective values shown in tables 2 and 3, a company may decide to change the priority of resources (time and cost) in order to get a better payoff value.

4. DESIGNING A MODEL SELECTION SYSTEM

Based on the study in sections 3, we present a general model selection system in this section. The model selection system not only will select the best model for a problem given the time and cost constraints, but also allows the user to do simulation of model selection with different time and cost constraints.

The model selection system consists of a user input unit, a preprocessing unit, an optimization unit and a report unit. These units are described as follows:

1. The user input unit: This unit takes the description of the problem and resources constraints (for time $T_{\max}$ and cost $C_{\max}$) from the user.
2. The preprocessing unit: This unit uses the description of the problem entered by the user in last step to decide available models for solving the problem.

Many MMSs include model extraction features that allow users to extract relevant models for a problem (Turban & Aronson 2001). We will not go into the detail of this step.

3. The optimization unit: We assume that models in the model base have the various constants and precision functions stated in section 2.4 available on line. With this configuration, we have a GA program ready for each model. The GA program takes the time and cost constraints entered by the user in step 1 as inputs, and performs the algorithm a couple of times, say 10, with different initial population to obtain the near optimal solution.

4. The report unit: The near optimal solutions from all competing models selected from step 2 will be computed and compared to one another, and the one with the highest objective value will be selected and recommended to the user.

Today's information industry is by and large dependent on the cooperation between industry players. There is no single player that can do all of the jobs required in an intelligent information system. The model selection system we propose is such an example that allows for the cooperation between different information players. For example, the preprocessing unit in step 2 and the optimization unit in step 3 may be provided by different players that do their job best in each field. Since there is no direct input or output from or to the users between these two steps, it is possible to modularize each of these two parts and link them together via distributed computing technology. Web services technology is the current prominent technology in distributed computing because it is based on the industry standards such as XML and HTTP (Gottschalk et al. 2002). If we package the optimization unit according to the web services standard, then the preprocessing unit will be able to call this optimization service via the web services technology. This modularization and packaging will not only provide a flexible way to implement the system, but also promote the usage of a model selection system.

5. CONCLUSIONS

In this paper, we propose a methodology for selecting the best model from alternative ones in the model base subsystem of a DSS. The contribution of this paper is two fold. First, we introduce a paradigm based on the time and cost factors to analyze the profitability of a model solution. Adding the cost variables in the paradigm makes the analysis more complex, but we can represent the real world situation much better now. Secondly, we use a robust algorithm (GA) to solve the difficult optimization problem resulted from the analysis of model selection. This work can be considered as a novel study in the meta-data area of a MMS and its results extend previous work on MMS from creation, storage and retrieval of models to the selection of the best model.

Reference

Bonczek, R.H., C.W. Holsapple, and A.B. Whinston, "A Generalized Decision Support System Using Predicate Calculus and Network Data Base Management," *Operations Research*, Vol. 29, No. 2, 1981, pp. 263-281.

Koza, J.R., F.H. Bennett III, D. Andre, and M.A. Keane, Genetic programming III, Mogan Kauffman, 1999.

Lee, C.C., Model Selection for Real-time Decision Support Systems, Ph. D. Dissertation, Nation Sun Yat-sen University, 2002, Taiwan, ROC.

Liang, T.P. and C.V. Jones, "Meta-Design Consideration in Developing Model Management Systems," *Decision Sciences*, Vol. 19, 1988, pp. 72-92.

Liang, T.P., "Development of a Knowledge-Based Model Management Systems," *Operations Research*, Vol. 36, No. 6, 1988, pp. 849-863.

Mayer, M.K., "Future Trends in Model Management Systems: Parallel and Distributed Extensions," *Decision Support Systems*, Vol. 22, 1998, pp. 325-335.

Mészáros, Cs. and T. Rapcsák, "On Sensitivity Analysis for a Class of Decision Systems," *Decision Support Systems*, 16, 1996, pp. 231-240.

Ringuest, J.L., "Lp-metric Sensitivity Analysis for Single and Multi-attribute Decision Analysis," *European Journal of Operational Research*, Vol. 98, 1997, pp. 563-570.

Roy, J., "Recent Trends in Logistics and the Needs for Real-Time Decision Tools in the Trucking Industry," *Proceedings of 34th Annual Haiwaii International Conference on System Sciences,* 2000.

Snowling, S.D. and J.R. Kramer, "Evaluating Modeling Uncertainty for Model

Trebilcoke, B., "Welcome to e-World," Supply Chain Yearbook 2001, *Purchasing*, Boston, 2000, pp. 87-98.

Turban, E. and J.E. Aronson, Decision Support Systems and Intelligent Systems, 6th Ed., Prentice Hall, 2001.

Gottschalk, K., Graham, S., Kreger, H. and Snell, J., Introduction To Web Services Architecture, *IBM System Journal*, Vol. 41, No 2, pp. 170-177, 2002.