Systematic Evaluation of Design Decisions in CBR Systems

Juan Carlos Santamaría and Ashwin Ram
College of Computing
Georgia Institute of Technology
Atlanta, Georgia 30332-0280
{carlos,ashwin}@cc.gatech.edu

Abstract

Two important goals in the evaluation of an AI theory or model are to assess the merit of the design decisions in the performance of an implemented computer system and to analyze the impact in the performance when the system faces problem domains with different characteristics. This is particularly difficult in case-based reasoning systems because such systems are typically very complex, as are the tasks and domains in which they operate. We present a methodology for the evaluation of case-based reasoning systems through systematic empirical experimentation over a range of system configurations and environmental conditions, coupled with rigorous statistical analysis of the results of the experiments. This methodology enables us to understand the behavior of the system in terms of the theory and design of the computational model, to select the best system configuration for a given domain, and to predict how the system will behave in response to changing domain and problem characteristics. A case study of a multistrategy case-based and reinforcement learning system which performs autonomous robotic navigation is presented as an example.

Introduction

Two important goals in the evaluation of an AI theory or model are to assess the merit of the design decisions in the performance of an implemented computer system and to analyze the impact in the performance when the system faces problem domains with different characteristics (Kibler & Langley, 1988; Cohen & Howe, 1989; Aha, 1992). Achieving these objectives enables us to understand the behavior of the system in terms of the theory and design of the computational model, select the best system configuration for a given domain, and predict how the system will behave in case the characteristics of the domain changes.

Two important characteristics of case-based reasoning (CBR) systems are that they are complex and the domains in which they operate are also complex. One result of this is that the behavior of a CBR system has many sources for variability which causes any performance measure defined to evaluate this behavior to have variability as well. This in turn makes it difficult to assess the significance of an observed behavior of the system in a specific situation. Similarly, due to the complexity of the system and problem domains, theoretical analysis of the system performance given alternative design decisions and domain characteristics, although desirable, is difficult in many cases (Kibler & Langley, 1988; but see Francis & Ram, 1993). However, straightforward performance curves that show how the performance of a system improves over time are not good enough. Although these curves show that the performance improves, they do not provide useful information about why the system works or how the design decisions affect the behavior of the system. Ablation studies can be used to analyze the impact of different system's modules in the performance of the system (Cohen & Howe, 1988; Kibler & Langley, 1988). In such studies, one or more system modules are removed or deactivated to analyze how the performance of the system changes. Although these studies do provide some information about the merit of different modules in the performance of the system, they are based on extreme operating conditions that are often impractical (i.e., one or more modules are set to be either active or inactive). Moreover, design decisions often deal with allocating certain amount of resources to different modules. Due to their nature, ablation studies can only deal with all-or-nothing resource allocation, disabling the possibility of deciding what would be the optimal amount of resources to allocate to each module.

As an alternative to these approaches, we propose the use of statistical tools to analyze the change in the performance of the system in terms of changes in design decisions and domain characteristics. In such an analysis, the system is evaluated through systematic experiments designed to filter out undesirable sources of variability. The results of the experiments are then analyzed using statistical tools to identify the sources of variability in the behavior of the system and their significance.

This paper presents an evaluation methodology based on well known statistical tools that can be used to explicitly analyze the merits of the design decisions in the performance of a system and predict the impact of this performance when the domain characteristics change. The methodology consists of designing experiments to carefully control the variability in the behavior of the

Table 1: Systematic evaluation methodology.

1.	Experimental Design and Data Collection.
2.	Model Construction.
3.	Model Validation.
4.	Robustness Analysis.

system and to obtain data that can be used to construct a mathematical model that relates the change in the performance of the system with the alternative design decisions and domain characteristics. Such models can be used to select the best system configuration for a given domain and to predict the behavior of the system when the domain characteristics change.

This paper is organized as follows. We begin by describing the proposed methodology. We then present an application of the proposed methodology to SINS, a case-based system that performs autonomous robotic navigation. We conclude by summarizing the implications of the methodology for the evaluation of CBR systems.

Evaluation Methodology

The proposed evaluation method is shown in Table 1. This method can be used to explicitly analyze the merits of the design decisions and the generality of the system. It consists of four phases: experimental design and data collection, model construction, model validation, and robustness analysis. During the experimental design phase, the *factors* that may influence the performance of the system are identified. These factors are usually classified in two broad groups: *design decisions* and *domain characteristics*. Experiments are designed to measure the performance of the system while systematically varying the factors. These experiments are executed and the data is collected. During the model construction phase, empirical *models* that relate the design decisions, domain characteristics, and the performance of the system are constructed. In the model validation phase, the assumptions identified during the model construction phase are verified. In this manner, the models can be used to state valid conclusions about the relationship between the system's performance and the factors (i.e., design decisions and domain characteristics). Finally, during the robustness analysis phase, the system is tested under different alternatives for the factors to assess the generality of the results. The following sections discuss each of these phases in more detail.

Experiment Design and Data Collection

Case-based reasoning systems are typically complex in nature and their performance depends on several factors. These factors can broadly be classified into two categories: design decisions and domain characteristics. Design decision factors are related to the configuration of the system and often deal with allocating resources to different modules within the system. Domain characteristic factors are related to problem description and are used to categorize problems in the domain. To understand and optimize the performance of the system, it is necessary to assess the role of each factor in the system's overall behavior. During the first phase of evaluation, such factors are identified and experiments are designed to measure the system's performance for different alternatives or *levels* for each factor. A representative sample of systems and problem instances is selected, each one with a different set of alternatives along each factor. An experiment consists of measuring the performance of each system executing on each of the problem instances. Thus, an experiment requires more than a single run; it requires several runs carried out under different conditions (i.e., different configurations of the case-based reasoning system, different environmental configurations, different levels of problem "difficulty", etc.). In this way, it is possible to apply statistical techniques to decide not only which factors have influence on the behavior of the system and under what circumstances, but also to what extent. This information can be used both to explain why the system worked, as well as to select the best system configuration for a given problem domain.

While designing the experiments, it is important to reduce unwanted sources of variability in the system's performance across runs. It is also desirable to construct an empirical model that can explain differences in performance based solely on differences between alternative factors. To accomplish this, the experiments should either *balance* out the runs along the factors (i.e., to run all system configurations on problems instances with all levels of difficulty) or *block* out the runs along a specific factor (i.e., to run all system configurations on problem instances with only one level of difficulty). The choice about when to balance or to block a specific factor is made by trading off the cost of running experiments against the range of applicability of the results of the empirical model to be constructed. A model is applicable only to the range of problem instances from which it was constructed. Increasing the range of problem instances increases the range in which the model is applicable but also increases the number of experiments needed because each system must run under problem instances that represent the entire range.

Due to the fact that factors are often grouped by design decisions and domain characteristics, one practical way to design the experiments is to balance out all the factors related to design decisions and to block out all the factors related to domain characteristics. In this way, a detailed analysis of the merits of the design decisions under specific but representative problem instances can be obtained. Such analysis allows the selection of the best levels along the design decisions so that the system's performance is optimum when working under the representative problem instances. Then, during the robustness analysis phase, the generality of the best system configuration can be studied across different levels of domain characteristic factors. An approach similar to the robustness analysis phase is described by Aha (1992). He proposes an evaluation methodology designed to understand the effect of different domain characteristics in the performance of learning systems and to derive rules that designers can use to decide when to generalize the results obtained from case studies. In contrast, our methodology is designed to understand the effects of the design decisions in the performance of the system, to determine if the results are significant, and, furthermore to analyze under what domain characteristics the evaluation study remains valid.

Model Construction

After the experiments are run and the data collected, a mathematical model is constructed to fit and explain the results. Models that relate system performance and relevant factors (i.e., design decisions and domain characteristics) are useful because they provide information about how each factor influences the performance of the system. Such models can serve many purposes, such as predicting what the performance of the system would be under certain preselected conditions, and selecting the optimal levels of system parameters to configure a system for specific situations.

Due to the complexity of case-based reasoning systems, theoretical models that relate system performance and relevant factors are difficult to construct. Instead, the data collected during experiments can be used to infer an empirical model. Empirical models are mathematical expressions based on experimental data and can be constructed using statistical estimation techniques. The basic idea when constructing a model is to assume that there exists a functional relationship between system performance and the relevant factors. The model is a mathematical expression of this relationship.

An example of a linear empirical model is shown in Equation 1.

$$\mathrm{Y}_i = \beta_0 + \beta_1 \mathrm{X}_{1i} + \cdots + \beta_k \mathrm{X}_{ki} + \varepsilon_i \qquad (1)$$

The results of $i = 1, \ldots, n$ experimental runs are assumed to follow the relationship expressed in the equation. In the model, Y_i represents the dependent variable or observed performance of the system for each of the n runs, and the X_{ji} represent the independent variable or alternative values of each of the $j = 1, \ldots, k$ factors for each of the n runs. The value ε_i represent the *residual* or error incurred by the model in estimating the observed value Y_i given the values of the X_{ji} for each of the runs. Inferential statistical techniques are used to estimate the values of the β coefficients for the given sample (e.g., Least Squared Error Estimation).

The linear regression model in Equation 1 is very general and can incorporate a wide range of smooth functional relations. For example, ablation studies analyze the partial increment/decrement in the system's performance with the addition/elimination of a system component (Cohen & Howe, 1988). Such analysis can be performed using a regression model in which indicator variables can take on the values 1 or 0 to indicate whether a system component is present or not. A linear regression model can also be used with continuous valued parameters, such as amount of memory. Finally, quadratic terms or other functional forms can be incorporated into the model because the only restriction is that it must be linear in the β coefficients. Once the model is created, the best set of parameter values can be selected to optimize the performance of the system. When smooth functional relations do not apply, other models may also be used; see, for example, one-way or two-way analisys of variance, or analysis of covariance (see, for example, Neter *et al*, 1989).

A common problem when constructing a model is selecting appropriate independent variables to use. One solution to this problem is to consider all the possible subsets of independent variables and select the best model according to a specific criteria. The most common criteria is to select the model with the best multiple coefficient of determination (R^2). This coefficient measures the ability of the model to explain the variability of the response variable (Y_i). The greater R^2 is, the better the model explains the variability of the response variable in terms of the variability of the independent variables.

Model Validation

Any model estimated using an inferential statistical technique relies on a set of assumptions. The validity of the model constructed depends on the extent in which these assumptions hold for a given sample of data. For example, there are two assumptions implied in Equation 1. First, the residuals are assumed to have zero mean and constant variance across samples. Second, they are assumed to be independent and normally distributed. When the assumptions do not hold, any conclusions derived from the model may not be valid. Deviations from the assumption of the residuals having constant variance might lead to overestimates in the ranges of parameter values. This in turn causes the model to be inaccurate. Small deviations from the assumption of the residuals being normally distributed does not create serious problems, but major departures are of concern since the conclusions derived using the model might be incorrect.

To verify qualitatively that the residuals have constant variance, a plot of the residuals against each independent variable and against the fitted response variable is used. A normal probability plot is commonly used to verify the normality distribution assumption of the residuals.

Robustness Analysis

In the final phase of the methodology, alternative levels for the factors are tried and verified against the model. As suggested in the experimental design phase, the experiments in this phase should focus on the factors that are associated with the domain characteristics. In this way it is possible to analyze the sensitivity of the best system configuration as obtained from the model across different domain characteristics, and to verify the generality of the case-based reasoning system across a range of problems.

A Case Study

This section describes a case study that is intended as an example of how to apply the methodology proposed in the section above. The case study is based on a detailed analysis of the evaluation of a case-based system that performs autonomous robotic navigation. In this case study, the objective of the evaluation is twofold: first, to find a model that describes the relationship between the system's configuration parameters and its performance as measured by a suitable metric; and second, to evaluate the robustness of the performance of the system under different environmental conditions. The first objective enables us to understand the relationship between the configuration parameters and system performance, and to evaluate the merits of different design decisions in the behavior of the system, and to verify that the performance of the system will not deteriorate with large amounts of experience. Moreover, a model that relates the performance metric with the configuration parameters and amount of experience or *experience level* is also useful because it enables us to pick the best system configuration parameter for a given situation and obtain optimal performance from the system. The second objective, evaluation of the robustness of the system when performing under different environmental conditions, is useful because it enable us to verify the generality of the system, i.e., whether it is likely that the results obtained will hold when the system runs under different environments.

The following subsections describe in more detail the system we used in this evaluation and each of the steps of the evaluation methodology.

System Description

SINS (Self-Improving Navigation System) is a case-based system that performs autonomous robotic navigation in unstructured terrains (for a detailed technical discussion of the system, see Ram & Santamaría, 1993a; for a discussion of the CBR aspects of the system, see Ram & Santamaría, 1993b). Autonomous robotic navigation is defined as the task of autonomously and safely moving a robot from a source point to a destination point in an obstacle-ridden terrain. SINS uses a schema-based reactive control module for robot navigation, coupled with a multistrategy case-based and reinforcement learning module for on-line adaptation and learning during task performance. It is difficult to evaluate a system such as SINS because its behavior is the result of many factors interacting with each other and because it is designed to work under unstructured terrains. Also, some modules in the architecture perform random actions under certain conditions (for example, to explore). This causes the evaluation to be even more difficult because random actions increase the variability in the behavior of the system. Thus, as discussed in the introduction of this paper, measuring the performance of the system during a single run or performing ablation studies does not accomplish the objectives of a systematic evaluation, which are to analyze the impact of the design decisions and domain characteristics in the performance of the system and to select the best configuration parameters. As discussed below, a systematic statistical evaluation can be used to accomplish these objectives.

Briefly, SINS consists of a navigation module, which uses a schema-based reactive control method (Arkin, 1989), and an on-line adaptation and learning module, which uses case-based reasoning and reinforcement learning methods (Ram & Santamaría, 1993). The navigation module is responsible for moving the robot through the terrain from the starting location to the desired goal location while avoiding obstacles along the way. A set of control parameters can be used to change the behavior of the navigation module. The adaptation and learning module is responsible for learning control parameters to change the behavior of the navigation module in such a way that the performance of the navigation task is improved. In particular, the adaptation and learning module constructs mappings from sensory input information to appropriate control parameters. These mappings are represented as "cases" that encapsulate the system's navigational experiences.

SINS captures its experiences and stores them as cases. A case represents continuous sensory inputs and associated motor schema control parameters over a time window, and recommends appropriate control parameters to use in different situations. As the system gathers more experiences, it can create new cases by allocating unused memory or it can modify previous cases by modifying their content or by increasing their time windows. Several parameters affect the behavior and performance of SINS; in this case study, we focus on two such parameters in the case-based reasoning component. These two parameters define the maximum amount of memory the system can use to store its experiences: maximum number of cases (C) and maximum case size (S). When the maximums are reached, the system uses new experiences only to modify the content of the cases if it is appropriate to do so. Thus, different values of these parameters affect the performance of the system.

SINS navigates in randomly generated environments consisting of rectangular bounded worlds. Each environment contains circular obstacles, a start location, and a destination location. The position, number, and radius of the obstacles are randomly determined to create environments of varying amounts of *clutter*, defined as the ratio of free space to occupied space. 15% clutter corresponds to relatively easy environments and 25% clutter to difficult environments.

In this evaluation, we will focus on how three factors influence the performance of SINS: maximum number of cases, maximum case size, and world clutter. The first two, maximum number of cases and maximum case size, belong to the design decision group. The third one, world clutter, belongs to the domain characteristic group. We will also consider how the experience level influences the performance of SINS and verify that the system indeed improves its performance as the experience level increases.

16

Experimental Design and Data Collection

As described earlier, the objective of this evaluation is to find an empirical model that describes the relationship between the system's configuration parameters and its performance as well as the conditions under such model is applicable. In this way, it will be possible to optimize the performance of the system by selecting the appropriate configuration parameters and to analyze the robustness of the system's performance when dealing under conditions that differ from the conditions in which the system was optimized.

To collect data for the evaluation analysis, we performed several runs on the system. A run consisted of placing the robot at the start location and letting it run until it reached the destination location. The data for the estimators was obtained after the system terminated each run. This was to ensure that we were consistently measuring the effect of learning across experiences rather than within a single experience (which is less significant on worlds of this size anyway).

We evaluated the performance of SINS using the median value of the time it takes to solve a world. The reason for this is that the median is a robust estimator of the mean and is not too sensitive to outliers. Outliers are common in schema-based reactive control since the system can get trapped in local minima points, resulting in a significant change in the behavior of the system. An experimental outcome consisted of measuring the time SINS takes to solve a world across five independent runs under the same conditions (i.e., same number of cases, case size, and level of experience, world clutterness) and reporting the median among the five runs as the response variable.

Two experiments were designed to satisfy the objectives of our evaluation. In the first experiment, we ran different systems under the same 15% cluttered world. Each system used different configuration parameters. In this way, we collected the data required to build a model that relates the system performance with the configuration parameters and amount of experience when dealing with a specific 15% cluttered world every time. In the second experiment, we ran the best system configuration, as determined by the model created during the first experiment, in a randomly generated 20% cluttered world. In this way, we could verify if the performance of the system holds when the domain characteristic changes. In this way we could balance out the effects of the configuration parameters and experience level and block out the effects of other factors such as world clutterness. The first experiment allows us to determine how the design decisions affect system performance (i.e., different systems under the same world or environment). The second experiment allows us to study how different domain characteristics affect system's performance (i.e., the same system under different environments).

Model Construction

As explained in the previous section, the performance of SINS is evaluated by estimating the median time to solve a world. Thus, the model that needs to be determined in the first experiment has the median time (T) as the response variable; the model relates T with the configuration parameters and amount of experience. We used the following regressors as independent variables: maximum number of cases (C), maximum case size (S), and amount of experience (E). We also considered additional regressors such as the quadratic terms C^2, S^2, and E^2 and the quadratic interactions CE, SE, and CS. The reason for considering all these factors is to allow for the possibility that interaction terms may explain variability in the response variable better than individual terms. Statistical analysis was used to reveal which of these terms are really significant and should be considered in the final model. Equation 2 shows the complete hypothetical model.

$$\begin{aligned} T \;=\;& \beta_0 + \beta_C C' + \beta_S S' + \beta_E E' + \beta_{CS} C'S' + \beta_{CE} C'E' + \\ & \beta_{SE} S'E' + \beta_{CC} C'^2 + \beta_{SS} S'^2 + \beta_{EE} E'^2 + \varepsilon \qquad (2) \end{aligned}$$

where: V' is the standardized[1] value of a variable V.

Assuming that the mathematical relationship between the response variable and the independent variables is "smooth", a second order polynomial expression of that relationship, such as the one proposed by the model, is a good approximation. Also, early experiences with the system showed that its behavior was related to the maximum number of allowable cases, maximum case size, and amount of experience. The quadratic terms for the maximum number of cases and maximum case size allowed for the possibility of utility problems and the interaction terms were included to allow for the possibility of a direct relationship between the response variable and the terms.[2]

An all-subsets regression analysis was performed to determine which of the terms in the model are really significant (i.e., which terms have influence in the response variable). In this analysis, all possible subsets of regressors are considered and a model is constructed using each subset. We measure the optimality of the model by its *adj-*R^2 which is the adjusted coefficient of multiple determination. This coefficient measures the ability of the model to explain changes in the response variable by changes in the regressors. Its range is between 0.0, which means that none of the variation in the response is explained by variation in the regressors, and 1.0 which means that all of the variation in the response is explained by variation in the regressors. Thus, the larger the *adj-*R^2 the more explicative is the model.

The best model obtained with the all-subset analysis corresponds to the one having all the regressors as independent variables[3] (*adj-*$R^2 = 0.796$, $F = 205.824$,

[1] Use of standardized values instead of the original values helps to reduce roundoff errors and other problems with multicollinearity between independent variables.

[2] Among the three interaction terms only CS has physical meaning. The interaction term CS is a direct measure of the total amount of memory available to the system. This is an example of a particularly difficult evaluation problem since different design decisions can influence each other under conditions of resource limitations.

[3] The F statistic is used to determine the significance of

Table 2: Model coefficients.

Coefficients	Value	Std. Error	P-value	95% C.I.
β_0	72.23	0.78	0.000	(70.70,73.77)
β_E	−11.92	0.34	0.000	(−12.58,−11.26)
β_C	−5.79	0.34	0.000	(−6.45,−5.13)
β_S	1.97	0.34	0.000	(1.31,2.63)
β_{EE}	2.33	0.38	0.000	(1.59,3.07)
β_{CC}	2.99	0.42	0.000	(2.16,3.82)
β_{SS}	−0.95	0.42	0.024	(−1.78,−0.12)
β_{CE}	−4.32	0.34	0.000	(−4.99,−3.66)
β_{SE}	−0.91	0.34	0.008	(−1.57,−0.24)
β_{CS}	0.74	0.34	0.028	(0.08,1.41)

P-value = 0.000). Table 2 shows the statistical results for each individual parameter in the model as well as the 95% confidence interval estimation of its real value.

Considering this model, the optimal system configuration parameters can be found using standard calculus techniques, i.e., by setting the first partial derivatives of the model with respect the relevant parameters to zero. Equations 3 and 4 shows the optimal values for C' and S' at a given level of experience E'.

$$\begin{aligned} C' &= \frac{2\beta_{SS}\beta_C - \beta_{CS}\beta_S}{\beta_{CS}^2 - 4\beta_{CC}\beta SS} + \frac{2\beta_{SS}\beta_{CE} - \beta_{CS}\beta_{SE}}{\beta_{CS}^2 - 4\beta_{CC}\beta SS}E' \\ &= 0.80 + 0.75E' \qquad (3) \\ S' &= \frac{2\beta_{SS}\beta_S - \beta_{CS}\beta_C}{\beta_{CS}^2 - 4\beta_{CC}\beta SS} + \frac{2\beta_{SS}\beta_{SE} - \beta_{CS}\beta_{CE}}{\beta_{CS}^2 - 4\beta_{CC}\beta SS}E' \\ &= 0.05 + 0.41E' \qquad (4) \end{aligned}$$

According to these equations the optimal parameter values change with the level of experience. This due to the interaction terms that exists among those variables. These equations can be used to determine the optimum configuration of the system for a given situation (an example is discussed below).

Model Validation

There are two assumptions that must be verified before accepting the proposed model as a valid model: The residuals have zero mean and constant variance, and the residuals have normal distribution. The LSE technique relies on these assumptions; since the model coefficients were calculated using this technique we must verify if these assumptions hold.

A scatter plot of the residuals against the fitted response was used to diagnose changes in variance and a normal probability plot of the residuals can be used to verify the normality distribution of the residuals. The scatter plot showed a constant band of residuals along the horizontal axis. Thus, this plot indicates that the variability of the residuals is constant along the fitted values of the response variable (i.e., median time). When the variability of the residuals is not constant, the band tends to narrow or widen along the horizontal axis. The normal probability plot showed a straight line that crosses the origin. This indicates that the residuals are indeed normal. When the distribution of the residuals is not normal, deviations from the straight line can be observed.

Table 3: Model coefficients.

Coefficients	Value	Std. Error	P-value	95% C.I.
α_0	80.2	0.71	0.000	(78.57,81.47)
α_E	−2.86	0.48	0.000	(−3.84,−1.87)
α_{EE}	2.53	0.55	0.000	(1.41,3.65)

Since the two assumptions, residuals with zero mean and constant variance and residuals having normal distribution hold, the model can be considered valid.

Robustness Analysis

A second experiment was designed to evaluate the generality of the SINS approach. In this experiment, we evaluate the same system performing under different environments. The data for the experiment was collected in the same manner as the first experiment, the only difference being that the robot solved a fixed randomly-generated 20%-cluttered world in every run. The configuration parameters for the system were selected using the model constructed in the first experiment and to optimize the performance of the system around an experience level E equal to 20 (i.e., $E' = 0.52$). Subject to these conditions, the system was configured using 43 maximum cases ($C' = 1.19$) of size 11 ($S' = 0.26$).

As in the first experiment, the model that needs to be determined has the median time (T) as the response variable. But, in this case, the model relates the response variable with the amount of experience only since the other factors are constant. In this way, if such a model is found to be significant (i.e., the model shows that the amount of experience is related to the response variable) we can conclude that the system still learns under changing environmental conditions. The coefficient derived from this model can be compared with the coefficients derived from the previous model. If a significant difference is detected, we can conclude that changing the world clutterness from 15% to 20% affects the learning performance. Equation 5 shows the complete hypothetical model for the second experiment. This model is a simplification of the model in equation 2 where only the experience level (E') is included as a regressor. Table 3 shows the statistical results for each individual parameter in the model as well as the 95% confidence interval estimation of its value.

$$\mathrm{T} = \alpha_0 - \alpha_{\mathrm{E}}\mathrm{E}' + \alpha_{\mathrm{EE}}\mathrm{E}'^2 + \varepsilon \qquad (5)$$

As the inferred model shows, a bigger intercept value is obtained which means that the system indeed needs more time to solve a 20% cluttered world. Also, the increased world clutter has a big influence in the rate of

the regression. The P-value is the probability determined by F; the lower this value the better the result, since the significance of the regression is (1 − P-value)%.

learning (α_E), which is reduced from −17.30 to −2.86. This means that more experience level does not improve the performance (reduce the mean time) as fast as in 15% cluttered world. The acceleration of the learning rate (α_{EE}) does not seem to be influenced by the change of world clutter (i.e., it is in the 95% confident interval of β_{EE}).

Evaluation Conclusions

The performance of SINS is very complex and depends not only on simple terms but also on their interactions. The evaluation shows that the median time the system takes to solve a 15% cluttered world decreases mainly as the experience level increases. Increasing the maximum number of cases also improves the performance, but a positive coefficient in its quadratic term may deteriorate the performance for big values. On the other hand, the maximum case size has a positive linear coefficient and a negative quadratic coefficient which indicate that large cases may improve performance as compared to small cases. Negative interaction coefficients indicate that for bigger values of maximum number of cases and cases size, the system requires more experiences to start improving its performance. Intuitively, this is to be expected since the more space is available to store regularities, the more experience level is required to construct reliable regularities. Finally, the performance of SINS is influenced by the world clutter, the learning rate being the factor subject to the greatest influence.

In summary, the evaluation was useful to verify and understand several aspects of SINS. In particular:

- The evaluation showed that SINS does improve its performance significantly with experiences (Tbl. 2).
- The evaluation showed that the performance of the system in a 15% cluttered world depends on alternative design decisions, as well as on interactions among them (Eq. 2 and Tbl. 2).
- The evaluation showed the best way to configure SINS in a 15% cluttered world for a prespecified level of experience (Eqs. 3 and 4).
- The evaluation showed how a change in the environment characteristics, namely clutter, affected the performance of SINS (Eq. 5).
- The evaluation showed that using the proposed factors (C, S, and E) and their interactions the empirical model can only account for 79.8% (i.e., R^2=0.798) of the variability in the performance of the system. Part of the remaining 21.2% could be explained by introducing more factors or by changing the functional forms of the terms in Eq. 2; the rest of the variation in performance is due to the randomness in the system.

Conclusions

Case-based reasoning systems are typically very complex, and the behavior and performance characteristics of such systems are the result of many interacting factors that originate from the many design decisions that go into building them. Additionally, the tasks, domains, and problems that case-based reasoning systems have typically addressed are also very complex and have a significant influence on the behavior and performance of the system. A good evaluation must show not only that a system is performing well; it should also inform us about the significance of the performance of the system under various conditions and provide insight about how the design decisions influence its performance. This allows the researcher to analyze the theory or computational model based on empirical experiments with the computer program, and the system designer and user to optimize the configuration of the computer program for a given situation of interest. A better understanding of the behavior of the system across domain characteristics also allows the designers to predict under what conditions the system will perform adequately.

Acknowledgements

The authors wish to thank Mary Dowling for all her useful comments and suggestions during the case study statistical analysis and Todd Griffith and Anthony Francis for their comments on an earlier draft of this paper.

References

Aha, D.W. 1992. Generalizing from Case Studies: A Case Study. In *Proc. of the Ninth International Conference of Machine Learning*, 1–10, Aberdeen, Scotland.

Arkin, R.C. 1989. Motor schema-based mobile robot navigation. *International Journal of Robotics Research*, (84):92–112.

Bareiss, R. 1989. The Experimental Evaluation of A Case-Based Learning Apprentice. In *Proc. of the Case-Based Reasoning Workshop*, 162–167. Florida.

Cohen, P.R. 1989. Evaluation and Case-Based Reasoning. In *Proc. of the Case-Based Reasoning Workshop*, 168–172. Florida.

Cohen, P.R. & Howe, A.E. 1988. How evaluation guides AI research. *AI Magazine*, 9(4):35–43.

Cohen, P.R. & Howe, A.E. 1989. Towards AI research Methodology: Three Case Studies in Evaluation. *IEEE Transactions on Systems, Man, and Cybernetics*, 19(3):634–646.

Francis, A.G. & Ram, A. 1993. Computational Models of the Utility Problem and their Application to Case-Based Reasoning. In *Proc. of the Third International Workshop on Knowledge Compilation and Speedup Learning*, 48–55. Amherst, Massachusetts.

Kibler D. & Langley, P. 1988. Machine Learning as an Experimental Science. In *Proc. of the Third European Working Session on Learning*, 81–92, Glasgow, UK.

Neter, J., Wasserman, W., Kutner, M.H. 1989. *Applied Regression Models.*

Ram, A. & Santamaría, J.C. 1993a. Multistrategy Learning in Reactive Control Systems for Autonomous Robotic Navigation. *Informatica*, 17(4):347–369.

Ram, A. & Santamaría, J.C. 1993b. Continuous Case-Based Reasoning. In *Proc. of the AAAI Workshop on Case-Based Reasoning*, 86–93. Washington, DC.