MACHINE LEARNING TECHNIQUES IN USABILITY-EVALUATION QUESTIONNAIRE SYSTEMS

Elena García[1], Miguel A. Sicilia[2], León A. González[1] and José R. Hilera[1]

[1]Computer Science Department
University of Alcalá. Ctra. Barcelona km. 33.600
28871 Alcalá de Henares. Madrid (Spain)
{elena.garciab, leon.gonzalez, jose.hilera@uah.es}
Phone +34 91 885 66 63. Fax: +34 91 885 66 46

[2]DEI Laboratory. Computer Science Department
Carlos III University. Av. Universidad, 30
28911 Leganés. Madrid (Spain)
{msicilia@inf.uc3m.es}
Phone: +34 91 624 91 04

ABSTRACT

Questionnaires are widely used instruments in usability evaluation, but their correct construction and administration may increase evaluation costs in terms of time and resources. In order to decrease them, evaluators often use "quick and dirty" questionnaires and, in some cases, obtain biased results. These costs could be reduced if information extracted from a baseline of past evaluations were used, applying the results of machine learning or mining process. In this work, we present how different tasks in a questionnaire life cycle can be more easily developed using machine learning algorithms, concretely, tasks related with the questionnaire design and the participants recruitment.

1 INTRODUCTION

Nowadays, the usability of interfaces is being considered a factor of increasing importance in application development. Usability [ISO 1993] can be defined as the effectiveness, efficiency, and satisfaction with which specified users achieve specified goals in particular environments, and it's taken into account in all phases of the development life cycle [Mayhew 1998], including usability evaluation in different process's stages.

There are three general types of methods to carry out usability evaluation: testing, inspection and inquiry, which comprise different techniques, like user testing [Dumas 1999], heuristic methods [Nielsen 1994] or questionnaires [Oppenheim 1992]. In this work, we focus on usability evaluation through questionnaires, which are used to obtain information about

users' likes, dislikes, needs, and understanding of the system by asking them about some concrete interface's aspects.

In order to obtain reliable results using questionnaires, statistical techniques are usually needed [Trochim 1999]. Test designers have to obtain validity and reliability measures and they also have to apply some kind of statistical inference, like analysis of variance, hypothesis testing or interval estimates of the variance. Experimentally obtained data is the unique information source that these statistic models take into account, and results inferred from a questionnaire don't supply useful data for future questionnaire construction, administration and analysis. All these factors increase costs when questionnaires are used for usability evaluation [Mayhew 1994], so, in many cases, collectors use 'quick and dirty' questionnaires (non-validated and often unreliable) and use only descriptive statistic methods (e.g. means and standard deviations) instead of statistic inference (as pointed out in ch. 20 of [Dumas 1999]).

In consequence, if a 'questionnaire system' that uses current statistical techniques were developed, the system would not exhibit any kind of learning behaviour – even in the case of having collected a huge amount of useful data –, since current analysis techniques don't take into account neither *a priori* knowledge (acquired from usability or domain experts), nor information extracted from the baseline of past evaluations as a result of some sort of machine learning or data mining process.

The purpose of this work is to describe how some specific machine-learning algorithms can be used to obtain knowledge that will subsequently be used to implement some basic learning behaviours in a questionnaire design system. The rest of the paper is structured as follows: the next section describes the concrete tasks in a questionnaire life-cycle that can be more easily developed using machine learning processes and the algorithms applied to achieve this goal. Section 3 sketches an enhanced tool prototype that integrates learning capabilities in a questionnaire-based usability evaluation system, and, at last, some conclusions and future work are summarized in section 4.

2 LEARNING BEHAVIOURS IN QUESTIONNAIRE SYSTEMS

A system aimed at designing, distributing and analysing questionnaires for usability evaluation should model the elements which take part in the real task [García 2001a], like the questions and the questionnaires themselves, the testers who evaluate interfaces and the evaluated systems or part of them. On the basis of the information that represents these aspects, the system may exhibit diverse learning behaviours. In this section, we sketch how these behaviours can be acquired.

2.1 LEARNING ABOUT THE TESTERS

The selection of suitable testers for a questionnaire is a hard task that entails costs in the overall price of the evaluation technique, since it's difficult to find the most adequate ones [Ehrlich 1994]. Empirical studies have been carried out to justify the use of a tester database instead of hiring external services to collect them [Rohn 1993]. But the existence of a tester

database is not enough to guarantee a reliable result. In fact, even in the specific case that usability experts were used to evaluate an interface, it's important that the group of testers is not biased.

If tester profiles are stored in the system, clustering techniques can be used to obtain natural groups of evaluators. The attributes used to drive the clustering can be extracted by aggregating the results of questions that are related to each concrete evaluation criteria for each evaluator. For example, different Likert questionnaires may evaluate interface attributes like learnability, efficiency, effectiveness, control or attractiveness asking participants their opinion about this system features. As each question in a questionnaire is associated to one or more attributes of this list, we can obtain an aggregated numeric value for each participant and each evaluated attribute using the stored evaluation facts (question result for each participant and each evaluated object) and we can use as input data set the different aggregated values tuples, one tuple for each participant. The COBWEB algorithm [Fisher 1987] can be used to obtain a number of candidate clusters, and the system automatically takes a balanced sample for new administrations.

2.2 LEARNING ABOUT QUESTIONS AND QUESTIONNAIRES

The reliability of a satisfaction questionnaire is directly related to the number of its items: the larger the number of items, the higher the reliability of the questionnaire [Nunnally, 1978]. However, a questionnaire with a large number of items takes a long time to complete. Therefore, it's interesting to try to reduce the number of questions so that more people would be willing to complete it. Questionnaire designers must decide and/or redefine the size of the set of questions in order to maintain the most suitable reliability/number of questions trade off.

In this context, the identification of strong dependencies between question responses can be helpful in the decision of which questions could be removed from the questionnaire. We say that a strong dependency exists between two questions q_i, $q_j \in Q$ -taken from the same questionnaire- if independently of the evaluated object and the participant, answers to both questions are equal the most of the times:

$$\forall t \in T, \forall o \in O, (f_i^{t,o} \approx f_j^{t,o})$$

where $f_i^{t,o}$ stands for the list of evaluation fact values obtained from tester *t* by the application of question *i* (in a given questionnaire) to an object *o* . This definition can be extended to more than two questions.

We have chosen *Apriori* algorithm [Agrawal 1994] for the extraction of association rules between questions. The application of the algorithm on answered questions requires a first step before the mining process (a pre-processing phase). We first must change the relational database format of our evaluation facts to one in which each application of a questionnaire is represented in a tuple with questions as attributes. For example, a tuple taken from the application of a Likert-scale questionnaire with seven questions may look as follows: (low, very-low, low, high, high, low, low). Note that order is important, that is, the nth attribute stores responses to the n^{th} question. Then we need to carry out a second transformation to

extract instances that can be directly taken as inputs by *Apriori*. For each tuple, we extract a set of 'boolean' tuples describing the different subsets of equal responses. For example, following the preceding example, responses one, three, six and seven would produce a tuple t1 = (true, false, true, false, false, true, true), and responses four and five would produce another tuple t2 = (false, false, false, true, true, false, false). So, a questionnaire with k questions would produce at most k/2 tuples of booleans (with the symbol / standing for integer division), and each of these tuples represents a positive fact about equality in subsets of the responses. Therefore, a table with boolean-valued attributes (q1, q2,…, qk) is given as input to *Apriori*. As the number of tuples can be greater than the number of questionnaire answers, the support parameter of the algorithm must be adjusted to the value: *1/(k/2)*. The algorithm produces a set of association rules in the form A $\Rightarrow$ B where A and B are sets of attribute/value pairs. Rules of arbitrary length are produced but we're only interested in those that are comprised only by 'true' values. That is, the rule q_1=true, q_2=false $\Rightarrow$ q_5=true has no meaning for our purposes (although it reflects a strong dependency in data). Therefore, we filter the output rule set to obtain a subset of rules in the form $q_i \Rightarrow q_j$ where q_x stands for q_x = true for any x.

Note also that we can't use directly the result of the data mining process since it may contain association rules that have nothing to do with redundancy, and therefore, questionnaire designer opinion is needed to assess the nature of the inferred rules, giving him/her the opportunity to discard or store the dependency as useful metadata about the questionnaire. Despite this limitation, the automatic discovery of dependencies in databases of evaluation facts helps usability experts in gaining insight about the aspects their questionnaires are attempting to evaluate.

2.3 Learning About the Evaluated Objects

Learning about past evaluations may enable the prediction of a system's usability in an automated way (we have restricted ourselves to the case of Web applications). This kind of evaluation is an effective complement to non-automated methods. Automated evaluation only considers internal attributes (see [Brajnik 2000]), that is, attributes that can be obtained by parsing the HTML code. We use the following quantitative measures [Ivory 2000], that can be computed automatically: Total number of links, page size in bytes, total number of images, total colours employed, reading complexity, and the number of text areas highlighted with bordered region, colour, rules or list.

Once we have found the measures of the page, we use a classifier to predict if it's usable or not. As measures are strictly numeric, we use the M5' algorithm [Wang 1997] to induct the model tree to classify new instances. This algorithm has been selected because model trees combine regression equations with regression trees, achieving lower average error values on the training data. The measures to train the classifier are obtained from pages that have already been classified by experts, so the classifier is provided with a training set for supervised learning.

3 SOME IMPLEMENTATION DETAILS

In order to consider and exploit knowledge extracted from past evaluations and to provide usability evaluators and questionnaire designers a useful way to make good use of this information, we are implementing a CASUE (Computer Aided Software Usability Engineering) tool that integrates the described learning behaviours with questionnaire design, questionnaire administration and response collection tasks. Main generic requirements that we consider in the construction of the tool are basically the following ones: (1) The tool should be designed to be extensible for today and tomorrow usability practices, so we are defining abstract data models for evaluation artefacts, concretely questionnaires (like the ones sketched in [García 2001b] and [García 2002]) and (2) the tool should enable the production of a baseline of usability facts that can be exploited by data analysis and knowledge management tools, so models are translated into conventional databases that are treated as operational data from which summaries and findings can be extracted.

The generic architecture for the integrated CASUE tool is mainly made up around the Evaluation Repository (ER), that represents the questionnaire and the other elements that take part in the evaluation facts and enable the application of knowledge extraction algorithms. Subsystems like Evaluation Performer (EP), Evaluation Miner (EM) and Evaluation Workbench (EW) are built around the ER. The EP is responsible for the automated delivery of evaluations to evaluators and the gathering of their results, by using some kind of delivery mechanism like the Web. The EW is a tool aimed at the efficient design of the evaluations that will be performed by the EP. Finally, the EM is a generic name for evaluation tracking and AI-based modules or data analysis tools that allow the findings extraction, which provide information to design efficient evaluations by means of EW subsystem tools. EM subsystem implements the machine learning processes using WEKA libraries [Witten 2000] and triggers them when they reach a predefined increase threshold in the volume of new available data or when the user requires it explicitly. In addition, some a priori knowledge can be exploited in the form of similarity relations between questions in different questionnaires to carry out rough predictions and compare assessments obtained through different questionnaires.

4 CONCLUSIONS AND FUTURE WORK

We have tested the techniques described here only in a medium size repository, and, under this condition, we have obtained the awaited results. However, we can guess some possible constraints in the application of some algorithms on actual data repositories. Specifically, the clustering approach requires that all evaluators had answered a high number of questions, and that they had evaluated all the criteria. Besides, as the values aggregated by criteria are not independent variables, it's possible that using Cobweb results were biased. We are studying the application of other schemes to substitute the current one.

In order to obtain an enough amount of data that enables a more thorough validation of our approach, we are currently performing a usability evaluation of the principal Spanish universities web portals. We're also studying to extend the study using of different types of

web sites in order to identify the most suitable quantitative measures that enable the automated classification of this other web (for example, e-commerce ones).

REFERENCES

Agrawal, R., Srikant, R. (1994) Fast Algorithms for mining association rules. *Proc. of International Conf. Very Large Databases (VLDB'94)*, Santiago, Chile.

Brajnik, G. (2000) Automatic Web Usability Evaluation: What Needs to be Done?, *Proc. of the Sixth Conf. on Human Factors and the Web*, Austin, United States.

Dumas, J. S., Redish, J. C. (1999) *A Practical Guide to Usability Testing*. Intellec.

Ehrlich, K., Rohn, J. A. (1994) Cost justification of usability engineering: a vendor's perspective. *Cost-justiying usability*. Eds: R. Bias, D. Mayhew. Academic Press.

Fisher, D. H. (1987) Knowledge Acquisition Via Incremental Conceptual Clustering. *Machine Learning, 2*.

García, E., Sicilia, M.A., Hilera, J. R., Gutiérrez, J.A. (2001) Extracting knowledge from usability evaluations. *Proc. of Interact 2001*, Tokyo, Japan.

García, E., Sicilia, M.A., Hilera, J. R., Gutiérrez, J.A. (2001) Computer-Aided Usability Evaluation: A Questionnaire Case Study. *Proc. of Panhellenic Conf. on Human Computer Interaction*, Patras, Greece.

Garcia, E., Sicilia, M. A., Hilera, J. R., Gutierrez, J. A. (2002) Extending Question & Test Learning Technology Specifications with Enhanced Questionnaire Models. *Proc. of the Int. Conf. on Information Technology Based Higher Education and Training 2002 ITHET'02*, Budapest, Hungary.

ISO TC 159 // SC 4 Technical committee (1993). *ISO 9241 Ergonomic requirements for office work with visual display terminals (VDTs) - Part 11: Guidance on usability*.

Ivory, M. Sinha, R., Hearst, M. (2000) Preliminary Findings on Quantitative Measures for Distinguishing Highly Rated Information-Centric Web Pages. *Proc of the Sixth Conf. on Human Factors and the Web*, Austin, United States.

Mayhew, D., Mantei, M. (1994) A basic framework for cost-justiying usability engineering. *Cost-Justiying usability*. Eds: R. Bias, D. Mayhew. Academic Press.

Mayhew, D. (1999) *The Usability Engineering Lifecycle*. Morgan Kaufmann.

Nielsen, J, and Mack, R. eds, (1994) *Usability Inspection Methods*. John Wiley and Sons.

Nunnally, J. C. (1978) *Psychometric Theory*. McGraw- Hill.

Oppenheim A. N. (1992) *Questionnaire Design, Interviewing and Attitude Measurement.* Pinter Pub Ltd.

Rohn, J. A. (1993) *Resource Investments for Usability Engineering Methods*. SunSoft Internal Study. SunSoft.

Trochim, W. (1999) *The Research Methods Knowledge Base*. Atomic Dog Publishing.

Wang, Y. and Witten, I.H. (1997) Induction of model trees for predicting continuous classes. *Proc. of the European Conf. on Machine Learning*, Prague.

Witten I. H., Frank E. (2000) *Data mining: Practical machine learning tools and techniques with Java implementations.* Morgan Kaufmann.