Simulation Modelling Practice and Theory 19 (2011) 47–55

Contents lists available at ScienceDirect

Simulation Modelling Practice and Theory

journal homepage: www.elsevier.com/locate/simpat

Partial user-supplied information and user modeling for improving QoS

Rodrigo Verschae [*], Mario Koeppen, Kaori Yoshida

Network Design Research Center, Kyushu Institute of Technology, 680-4, Kawazu, Iizuka, Fukuoka 820-8502, Japan

ARTICLE INFO

Article history:
Available online 6 July 2010

Keywords:
QoS
Quality of service
Partial answers
Prototype
User modeling
Network design

ABSTRACT

The incorporation of user-supplied information has become mandatory for the improvement of QoS in network systems. There is the question about accommodation of new users of a service, given that information about former users of a service is available. In the present work, we followed two approaches to derive information about new users in the network design and control processes, where both are based on prototype generation for the answers of former users to a QoS related questionnaire. In the first approach, attempts were made to map user attributes to prototypes. The second approach used a mapping from partial answers to a prototype. As a result, the first approach appeared to be infeasible, while the second showed good results. In the resulting trade-off between number of prototypes and classification accuracy, it is possible, for example, with 8 prototypes for around 1000 users to predict the answers of new users by using only 30% of the answers of former users, while reducing accuracy by only 13% at the same time.

1. Introduction

Everyone has already experienced searching for a missing item, for example, in one’s own home. Usually, to search for it, one will select a place, such as a particular drawer of a cupboard, and look there as a first attempt. If the item is there, it is fine. If not, the search has to go on, until the missing item eventually reappears. For storing it away again, it is generally a good advice to put it back to the same spot, where one was looking at first time.

This example of a simple daily experience shows a number of lessons. One is that it might be hard to make a good “cold start” assignment, like the assignment of items to places for storage, but it is always possible to improve the assignment over time, as soon as further “use cases” come in. Another lesson is that there is not always the need for a full comprehension of a retrieval scheme. If asked, in most cases, a subject might not be able to give a good reason for selecting the first place to look for the missing item. Nevertheless, it is the place most likely to be reproduced in a later rehearsal of the situation.

In computer networking [17], quality of service (QoS) [2,6,3] refers to the ability to provide different priority or to guarantee a certain level of performance to different applications, users, or data flows. For the corresponding design, evaluation, and adjustment of QoS, the situation is not much different from the example given above, thus similar lessons apply. Having a new user of a system, usually not much more is known about the user than that she belongs to the group of average users of a system. The best way to improve the QoS to that particular user is to monitor later encounters with the same user and gather experiences. Based on this, user-specificity can be reflected better and better by the service. The main point here is the need for some kind of predictions – completely unbiased at the beginning, but once a favourable prediction is made, there should be the possibility to store this information in a representation, which is suitable for the future improvement of the service not only to this user but also to other users.

[*] Corresponding author. Tel.: +81 948 29 7947.
E-mail address: rodrigo@verschae.org (R. Verschae).

1569-190X/$ - see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.simpat.2010.06.014

And for the other lesson: often today's services attempt to state conditional dependencies within a model of the user behaviour. A typical example is the "customers who bought commodity A usually also bought B, C,..." type of information, provided by many on-line merchandising systems. This is good and guides the customer, but it will never focus on the question *why* any of these customers ever did consider buying products based on some conditional probability.

In computer science, we find such facts often related to the concepts of (machine) learning and computational intelligence, or more general so-called data-driven approaches [5]. These are in contrary to e.g. rule-based approaches, despite some overlap. Rules try to express reasoning about the relation and mutual causality among data within a rigid, logic-based formal framework. The problem with rules is, simply said, that actually nothing guarantees their existence. And once their existence can be validated, the means for their acquisition in a real-world application might result in a heavy load of needed user information (commonly referred as "user fatigue"). Data-driven approaches rather attempt to instantiate generic formalizations (such as formula expressions, or the computation, represented by a neural network) to a real-world situation, without focusing on the suitability of such a formal approach and mainly basing their right for existence on "good performance", which is mostly expressed by a corresponding optimization task.

The goal of the research presented in this paper is to devise a data-driven approach to QoS to overcome the problem of user fatigue, i.e. to keep the amount of information acquired from a user at a minimum. We will introduce a new approach to user-preference modeling with the goal of improving QoS. The approach is a hybrid of supervised and unsupervised learning method. In this approach, the user is formally represented as a symbolic entity, which means a class label or a prototype in particular. This symbolic entity can be mapped onto features that can guide the specific design of a service. It should be noted that the meaning of "prototype" here is according to its use in machine learning: a functional representative or substitute of a class or category of objects. In addition, prototypes can be also used to simulate a larger number of users of a system than are actually available (like in an experiment), and it can be used to complete information about a user that is only partially given. The assignment of symbolic entity is achieved by unsupervised learning, while the mapping of symbolic entity to features is achieved by supervised learning.

For demonstrating the feasibility of the approach, we will provide a case study. The material of this study is a questionnaire about user preferences for accepting quality losses of several types of media during broadcasting. The fact that user mostly do not like such quality losses at all made the information from this questionnaire's outcome strongly biased. Also, the rather large number of questions posed the problem of user fatigue, i.e. asking too many questions to the users in order to characterize them in terms of their preferences should be avoided, otherwise the users may get tired or bored. The advantage of the proposed method lies in the fact that the evaluation of a dataset under the pre-dominance of a larger amount of negative scores becomes possible, and that user fatigue can be reduced as the missing information can be completed from the selected prototype.

The use of quality of service in network systems has been widely studied. Most works have focused on studying the QoS in IP network and Internet (e.g. [21,18]). In the case of IP networks, different terminologies have been used (see [8] for general overview). The models and theory of quality of service in Internet have focused on [18]:

- the use of network calculus to obtain deterministic performance guarantees,
- the design of architectures that can scale,
- the design of stateless cores,
- the study of statistical performance guarantees, and
- performance analysis in best effort context used for TCP performance modeling, differentiation of quality access without access control, and application control in the absence of network support.

In some cases, the research on QoS has focused on particular applications, such as web services [15], routing for supporting multimedia applications [19], or in optimization issues [14].

Although almost all efforts have been put into network modeling and design, the utilization of user data and user modeling has been started to be used in recent years. Taking the user and the user's preferences into account, and thus being able to adapt the system is important (see e.g. [12]). Resource allocation should take the state of the networks into account as well as the preferences of the users, when possible. Recently, the improvement of QoS based on user-supplied information, and following evaluation by machine learning, soft computing and other intelligent technologies has gained increasing interest [1,11,7]. For this procedure to be efficient, it needs to provide information (for example about the user's impression of a web service's quality) in such a way that it can be directly employed in the design and control of the communication networks providing these services. The typical problem in this context is that once there is a corpus of user-supplied information about QoS available, how to use this to handle new users of a system, or to handle the same users, but in newly appearing situations, and as efficiently as possible. Without stressing the cold-start problem too much (which simply cannot be solved), we consider the use of standard machine learning techniques to handle the new user problem, and to investigate the kind of information that is provided by the new user for the solution of this task in the best manner.

We contribute an investigation of two different approaches to the evaluation of a questionnaire about tolerance for technical media quality problems, related to media types and problem type, and which was performed via a web questionnaire [22]. The two approaches consider the introduction of new users by either learning about the new user's attributes (like age, education, preferences, etc.), or by acquiring a smaller amount of information (partial responses) from the user that already allows for closely relating the user to some users already known to the system (thus also reducing new user's fatigue, and less opening of user's private information). In both cases, users known to the system were represented by prototypes. The procedure to be designed is to map a new user to one of these prototypes, in order to allow further conclusions about a new user by equating her or him with an existing user profile.

In addition, the perspective of this research is that the answers to such a questionnaire could be directly employed in lower layers of the communication networks. The topic was the sharing of video quality problems among a group of users, depending on different utilities (represented by lower tolerance for quality problems) that users give, depending on the media type (movie, sports, news, etc.). However, we are not going to report on the means for distributing the different utilities within the end-to-end communication (e.g. by using a different streaming method), or via new routing concepts (for allocating higher traffic resources where the utility for the user is higher). This is topic of future investigations. Here, we focus on the provision of the information that can be used for such network design and control approaches, as a necessary first step.

The present work also extends a recent work on the evaluation of this questionnaire, where the focus was on the internal dependencies among the various tolerance assessments [13]. From this former work, it is already known that the answers in the questionnaire are dependent on each other on a per-user base, and independently of the media type. In the present work, this property already gives a good motivation to remarkably reduce the number of questions posed to the user. The contribution of this paper is a demonstration, by means of a case study, that the representation of users by such functional prototypes is feasible for the processing of user information towards achievement of QoS.

The paper is organized as follows: in Section 2 we will introduce the questionnaire to have the base for the following descriptions of the evaluations in Section 3. Then, Section 4 will present results achieved for the two approaches, and will be followed by a concluding section.

2. Subject and method

2.1. Dataset and acquisition

In February 2004, a questionnaire about the quality demands of users on multimedia content was run via the Internet for five days. The purpose of distributing the questionnaire was to find out about dependencies of user preferences for video quality losses from actual video contents. Besides personal information, the questionnaire contained 10 questions about the tolerance for multimedia content delivery problems for 9 different types of media content. The subjects were asked about 10 situations, in which quality losses of multimedia content delivery were apparent, and the subject could give a tolerance level from 1 to 4. These categories stands for: "1" being acceptable, "2" tolerative, "3" not tolerative, and "4" not acceptable. The 10 situations were described as follows:

1. Movie is no problem, but sound is interrupted some times.
(There may be no sound term.)
2. Movie is no problem, but sound is delayed.
(There may be movie–sound gap.)
3. Movie is no problem, but sound is with noise.
(It may be difficult to catch sound clearly.)
4. Movie is no problem, but text view instead of no sound.
(It is like subtitles with silence.)
5. Sound is no problem, but movie is interrupted some times.
(There may be blind term.)
6. Sound is no problem, but frame drops occur some times.
(Other frames except dropped frames are normal.)
7. Sound is no problem, but movie freezes.
(Movie stops some times.)
8. Sound is no problem, but movie gets blurred.
9. Sound and movie are no problem, but its start is delayed.
(There are a few seconds until start.)
10. There is no movie and no sound, only text is broadcasted.

The questionnaire had a finer granulation with respect of the type of media contents (drama/movie, sport, music, entertainment, documentary, leisure, news, anime, others). one thousand and fourteen subjects participated to the questionnaire, with a good distribution of gender, age, and kinds of network access. More details about the questionnaire, and its evaluation can be found in [22,13].

2.2. Past evaluations

In order to come up with a proper user model derived from the answers to a questionnaire, one aspect, which has not much been taken into account is the user's own attempt to keep a kind of coherence among her or his answers. The video quality questionnaire gives a tempting example, as the answer could also have been "4" all the time, which means that the user is not willing to differentiate among different ways of loosing quality of received video broadcasting in any cases. In fact, most users did not reply with "4" to all questions (but only a few did so). A recent study on this dataset was focusing on a new approach to, in a certain sense, measure this user's intra-dependence of questionnaire answers. The used method was proposed by Robert Hecht-Nielsen in his seminal work on confabulation cogency [10] as a model for human cognition and related foundation for computer-based cognition. This method was, as a demonstration example, applied to the generation (or better creation) of artificial news stories that, despite being artificial, are sound to the readers, as their generation was based on the evaluation of a huge backbone of newspaper articles and other English texts.

The relation of such a method to the idea of a user "inventing her- or himself" during the filling-in of a questionnaire might already become apparent from the comparison to such a demonstration example. In other words, it needs to evaluate the degree to which a questionnaire result is partially "invented" by the subject, in order to maintain an implicitly wanted coherence among the answers to her- or himself.

Cogency describes the degree to which the truth of a premise would make its conclusion more probable. Numerically, it is equivalent to a sampled conditional probability, but the way of concluding based on cogency is different from the corresponding Bayesian derivation of *a posterior* probability and their maximization. The evaluation here rather goes towards the joint cogency of a number of events, to make the conclusion most probable. The term "confabulation" now refers to a specific cognitive way (or better "trick") to handle joint cogency: it is assumed that humans organize their cognition in a manner that the dependence of joint cogency from cogency of subsets of premises besides of single premises is nearly constant. Thus, the joint cogency appears to be the product of simple cogencies times a nearly constant factor. This is an assumption that will only hold for subjects, and not, for example, for natural phenomena.

The cogency confabulation framework could be also applied to the video questionnaire dataset in order to represent the degree to which it is possible to "confabulate" user entries in the questionnaire from a smaller set of a user's entries. The issue is more deeply explored in [13]. For all cases between 1 and 10 (the number of different transmission errors, as this study ignored the media types), the corresponding cogencies were computed, and were used for artificially generating user questionnaire answers. The statistics of the simulated user answers, and the ones collected during the experiment, were compared, and a very strong correlation was found for the case of using 4 answers of the user. The interpretation, following the concept of cogency confabulation, would be that from just 4 of the 10 answers, the other 6 could be "invented" from a more general pattern (and notably also, no more complex interaction pattern was needed).

The further study on this dataset then was more focusing on visualizing this dependency among the user answers, as this has not been very obvious from a pure statistical evaluation of the dataset. An approach to visualization following the use of self-organizing maps (SOM) was presented in [16]. Here, the basic idea was to achieve a visualization of data different from the typical means of graphs, pie-charts, etc., but nevertheless to follow basic principles of human perception and cognition abilities. In fact, the method was to train a SOM on the complete set of data and recall it for subsets, according to the various quality and movie type categories. The distance of particular data items to the winning neuron was used to generate a distance map and present the SOM grid this way to the user. Then, the stronger relation between categories following the general pattern, or categories similar to each other but different from the general pattern became prevalent, and the procedure allowed for logical conclusions about the relation among the various categories, user attributes, and quality problems.

It should be mentioned that the visualization-by-SOM approach did only work after the removal of datasets with all evaluations being "3" and "4" only.

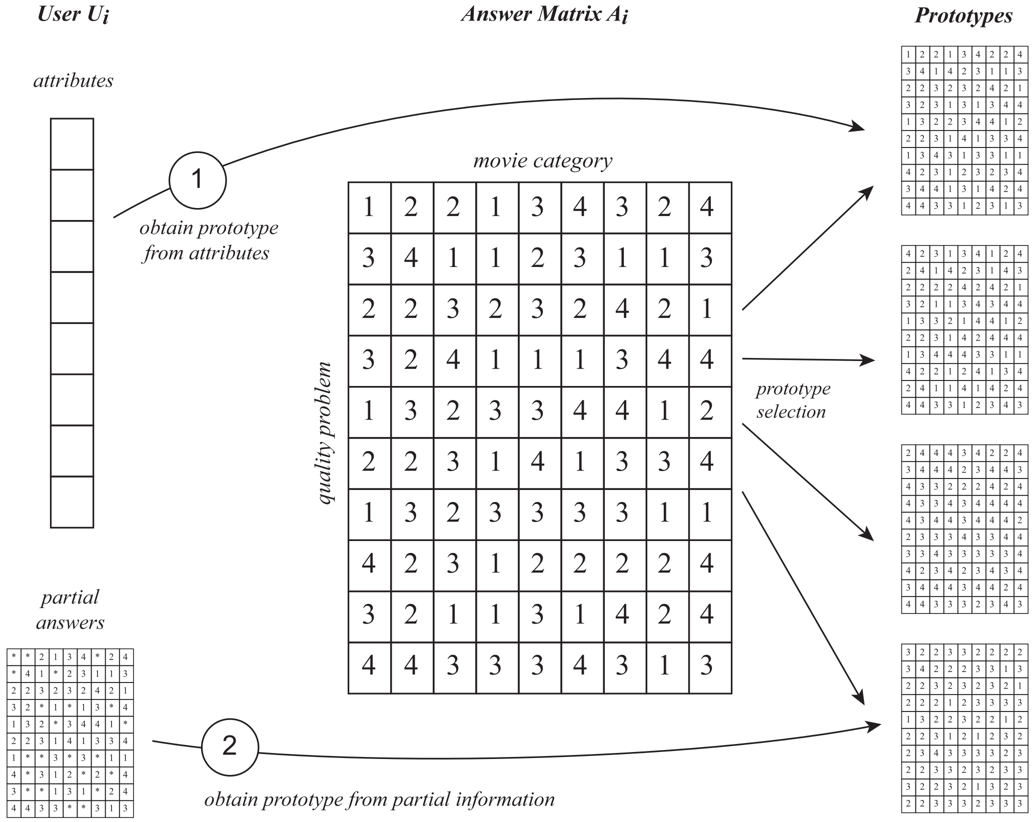
3. New user answer matrix prototype generation

3.1. System overview

The proposed system has two main modules: (1) generation of prototypes, and (2) mapping from user information (user attributes or partial questionnaire information) to prototypes. A prototype represents the user information needed for further processing, while the mapping allows to obtain that user information from incomplete or related information. Fig. 1 shows a diagram of system.

We are introducing a number of formal notations. The evaluation is performed on the answers given by each user. There are N users $U_1, U_2, \ldots, U_N$, and the answer of a user U_i is represented by an answer matrix $A_i = (a_{kl})$. For the quality demand questionnaire, the row index k is representing the type of media content and the column index l is representing the kind of delivery problem (which means that $k = 1, \ldots, 9$ for 9 media types and $l = 1, \ldots, 10$ for 10 kinds of delivery problems). The entries in the matrix are from an alphabet Z ("1","2","3","4" in case of the studied questionnaire). In addition to the answer matrix, there is a set of attributes D_i assigned to each user that formally represents the user profile as a n-tuple.

From the machine learning point of view, the goal of the QoS improvement is to provide a means for obtaining a feasible answer matrix for a new user of the system, based on the knowledge of the existing user answer matrix relations. Essentially there are two ways to accomplish such a goal, depending on the way how the knowledge about the new user is represented. The first approach is based on the user attributes, and the second approach is based on a partial setting of the answer matrix. Both approaches will be described in the next subsections.



1	2	2	1	3	4	3	2	4
3	4	1	1	2	3	1	1	3
2	2	3	2	3	2	4	2	1
3	2	4	1	1	1	3	4	4
1	3	2	3	3	4	4	1	2
2	2	3	1	4	1	3	3	4
1	3	2	3	3	3	3	1	1
4	2	3	1	2	2	2	2	4
3	2	1	1	3	1	4	2	4
4	4	3	3	3	4	3	1	3

Fig. 1. System diagram.

In both cases, we are representing the answer matrices by prototypes. A prototype answer matrix A^* is a matrix representing a group of answer matrices, and working as a proxy for corresponding evaluations and decisions, which normally depends on a user's answer matrix. The understanding is that such a prototype matrix is close (in the sense of a metric) to the group of answer matrices, which it is representing.

A cluster method C will represent the set of answer matrices as a union of pair-wise disjoint subsets. For convenience, all subset gets different symbols assigned, for example the numbers 1 to c, the number of clusters. In addition, each such subset (cluster) will be represented by a prototype (usually a matrix close to all answer matrices of a cluster). We may denote the prototype for the cluster with symbol k as A_k^*. Then, for the known users, the assignment of a symbol from the same symbol set is straightforward: it suffices to take the symbol assigned to the answer matrix of the user. But the question now is how to assign such a symbol, and prototype, to new users.

As mentioned before, we are considering two ways to do such an assignment, both based on available user information. They will be detailed next.

3.2. User assignment

In this approach, we are considering a classification of user's attributes, i.e. the generation of a mapping $Cl_A(D) = k$, where k is a symbol from the cluster symbol set and D is the attribute set for a user U. Then, a user (existing or new with the system) gets the prototype for cluster k as answer matrix assigned.

The creation of the classifier mapping will be based on supervised learning. A training set is provided from the existing user set. If the cluster method assigns the symbol k_i to the answer matrix A_i of user U_i, then the training data are a random selection from the set of pairs (D_i, k_i). The non-selected pairs will be used as test set T. The set $\{j | j \in T, Cl_A(D_j) = k_j\}$ represents all correctly classified user attribute data, and is used for quality assessment of the classifier.

3.3. Prototype selection from partial answers

The second approach is based on the idea that the user will not have to provide the complete answer matrix, but just a part of it. This means that we want to refer to information only, which can be more easily yielded and expanded from user's interaction with a service. Then there is the additional advantage of reducing user's fatigue, since less information needs to be available. Moreover, it does not require to give personal information of the user (in Section 3.2 user attributes are needed), which may be important for privacy issues.

Here we are also considering a mapping. The partial answer matrix is a matrix with entries from the same symbol set as before, and an additional wildcard answer "*". There is an embedded distance between an answer matrix (or a prototype) and a partial matrix, directly derived from restricting the distance evaluation to the non-wildcard positions and the corresponding positions in the answer matrix. This way, we can always assign a symbol $k = M(A)$ to any matrix A by taking the symbol of the closest prototype.

If we fix a set S_i of matrix index positions, and set all other entries to wildcards in the known set of user answer matrices (denoting such an answer matrix as A_i^-), and separate again into training set and test set, we can assign a quality measure for this approach as well, based on the size of the set $\left\{j | j \in T,\ M\left(A_j^-\right) = k_j\right\}$.

4. Results

4.1. Preliminaries

There are several ways of implementing the previously described modules. For simplicity we consider procedures based on two well known algorithms: *k*-means for clustering and naive-Bayes classifiers.

k-means is an agglomerative clustering algorithm that allows to partition data in groups of observations. The basic idea of *k*-means is to assign observations to groups based on the nearest mean distance to the groups. Here it is used to obtain a set of prototypes based on the available user data (questionnaire information). We consider different numbers of prototypes (2, 4, 8 and 16), where for each centroid of the obtained clusters a prototype will be assigned. Then, to each user with a known questionnaire answer $A_i = (a_{kl})$ the closest prototype is assigned. Therefore *k*-means is only used when building the system. We have used the implementation available in the software package R, which is based on the work presented in [9].

A naive-Bayes multi-class classifier is used to map user features to prototypes, with each prototype being represented as a different class. The naive-Bayes classifier assumes independence of the features (questions or attributes) and in practice it uses the MAP (maximum a posteriori) decision rule. As mentioned before, the features can be user attributes or partially completed questionnaire. We have used the naive-Bayes classifier implementation of Weka [20], which models each feature probability using Gaussian distributions.

4.2. Results and discussion

The used data consist of a questionnaire and user profile data obtained for 1014 users. A total of 90 questions were asked to the users, 10 questions per each of the 9 genres. There was also information about user profile (120 features), which included information going from things like if the user had a mobile phone or not (and by which company) to the area where they lived. Note that the results presented in the following were obtained using a 10-fold cross-validation procedure.

Two experiments were performed. First (Section 3.2), after obtaining a given set of prototypes, user profile features (120 attributes) were used to classify the prototype. This gave very bad results, with the classifiers having accuracies close to a random classifier. The same results were obtained using a naive-Bayes classifier, as well as others classifiers, such as C4.5 decision trees, SVM and Adaboost.

The closeness to a random classifier reveals that there is not much correlation between user attributes and user answers. In addition, the numerical representation of attributes, as it is needed for the classifier training, is very ambiguous. Numerous procedures can be devised to transform a symbolic attribute to a numerical domain. Together with the large number of attributes, the exploration of all possible combinations becomes infeasible.

The second experiment (Section 3.3) consisted of classifying the corresponding prototype using partial answers as features. This experiment was repeated several times using different number of prototypes (2, 4, 8 and 16) and different percentages of randomly selected features (from 10% to 100%). These results are presented in Table 1, and summarized in Figs. 2

Table 1
Percentage of features vs. number of prototypes (classes).

Percentage of features	Number of prototypes (classes)			
	2	4	8	16
10	84.50	69.29	57.65	48.56
20	89.04	76.99	65.84	56.56
30	90.62	78.58	69.79	59.92
40	93.19	83.02	73.84	66.73
50	93.97	86.67	77.59	72.36
60	94.97	87.66	81.24	75.32
70	95.66	88.85	81.24	76.90
80	96.24	90.03	82.13	76.90
90	97.04	90.23	81.93	79.27
100	97.04	89.83	82.52	77.59

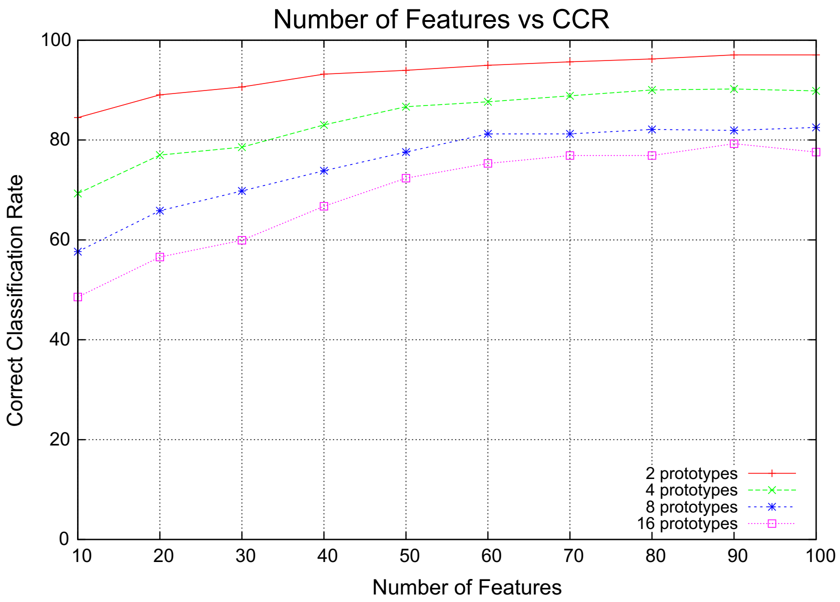


Fig. 2. Performance for different number of features.

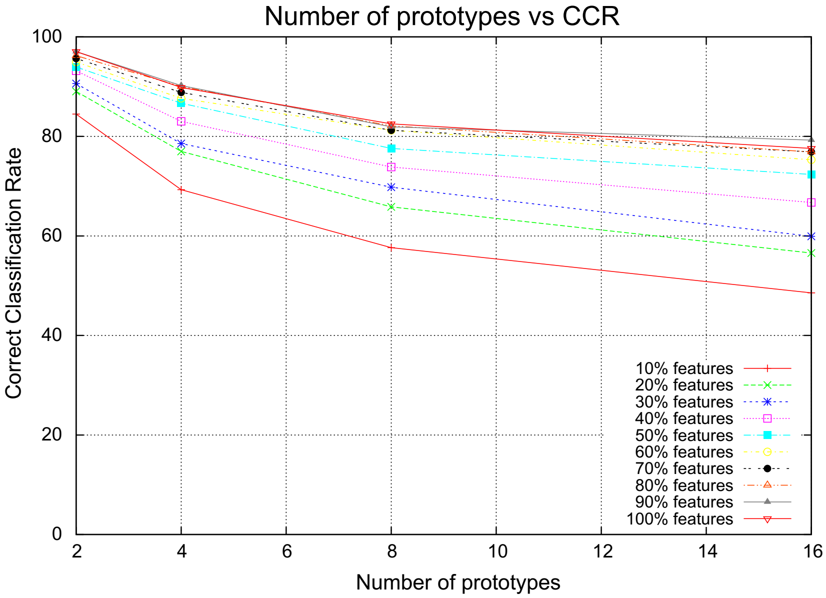


Fig. 3. Performance for different number of prototypes.

and 3. As it can be observed, using a larger number of features (answers) gives better results and using a smaller number of prototypes also gives better results. Reducing the number of questions to 1/2 reduces the accuracy of obtaining the right prototype by only about 5%, independent of the number of clusters.

It is important to recall the role of the prototypes and the influence of the number of prototypes being used. Each prototype represents a particular kind of user information. Having a larger number of prototypes would allow a finer personalization of the system, but it is unlikely that too many prototypes are needed, as there would be too many variables to be tuned on the QoS system. On the other hand, having few prototypes (2–3) may not be useful, as the diversity of users may not be well represented. If one takes, for example, 8 prototypes, and asking only 30% of the questions, reduces the accuracy by 13%, giving a classification accuracy of 69.79%, which we think corresponds to good enough trade-off between the number of prototypes and the accuracy of the system.

The results using different numbers of prototypes indicate that there are some clear "clusters" among the user answers. More interestingly, these results seem to be consistent with the results obtained in [13] where the same data was analysed and modeled using the ideas of confabulation theory. In that work it was shown that, using confabulation theory, it is possible to generate data with the same statistics as the input by using only 4 answers (out of 10). The fact that here only 30% of the questions reduces the accuracy by about 13% indicates that there is a kind of coherence among the user's answers (as observed in [13]) and here this coherence allows to use prototypes effectively and to reduce the number of questions, which in practice means that less data is needed (that in this case means diminishing user fatigue).

One final remark on the obtained results is that the prototypes obtained by *k*-means were well balanced, in the sense that the distribution of the assignment (smaller distance) of the user data (questionnaire) to the closest prototypes was mostly uniform, with each prototype (out of *k*) being assigned about $\frac{100}{k}\%$ of the data samples.

It is important to recall that in the performed analysis we considered simple models and methods for clustering (for obtaining the prototypes) and classification. The use of more advanced techniques, such as neural networks, boosting or support vector machines (SVMs) for classification, and spectral clustering, or fuzzy *c*-means for obtaining the prototypes, could help to obtain better results. For example, the use of a naive-Bayes classifier might not be the best choice in this case as the size of the training data is not large and the features may not be independent (as assumed in a naive-Bayes classifier). The use of SVM (with an appropriate kernel) or a Neural Network may overcome these problems. This analysis is out of the scope of this work and is one possible direction of future work.

4.3. Practical exploitation

The question that we want to target in this subsection is about a potential practical impact of such results. As already mentioned, the target is QoS, but under the umbrella of this acronym, several applicable techniques might be covered. At first, such results can be generally employed for improving recommendation systems. In this case, we have to consider two stages as parts of the proposed approach: to assign prototypes to a set of users, and to derive prototypes from partial knowledge about a user. The results for the first stage will allow for the global instantiation of the service, as it binds the set of use cases that will appear in practice. The second stage allows to conclude about a user without total knowledge, but also for incrementally adding new information (including the situation where the assigned prototype can even change, once having more reliable information from or about the user).

Apart from recommendation systems, the results are also of use for simulation of user behaviour, for example before putting a system into practice. Prototypes can serve information about particular users, user groups, and (also grouped) dependencies among particular user choices. In this sense, a prototype is more informative than just a symbolic class label, and is more efficient with regard to concluding than user attributes (whose selection is often more or less arbitrary). Then, prototypes, which are derived from real-world data, and also their features, can guide as a measure for correctness of a simulation.

In a related sense, the proposed approach also allows for quantifying the intra-dependence of user's answers to a questionnaire. This is a deviation from the common expectation that users will focus on answering the questions only. However, present analysis shows that a representative number of user answers can already be predicted from a subset of answers, and independently from the meaning of the questions.

Last but not least, there is also a prospective application to network routing. Within the current standard of network routing, and the corresponding protocols, all strictly following the paradigm of "network neutrality," such an application is not yet possible. The main methods used, e.g. for dropping packages in case of network congestion at a router, do not allow to differentiate the "semantic" of the data load of the package. However, it has been considered that putting more intelligence to the router might be the essential way of further improvements in network control [4]. In such a (future) scenario, the information about the user preferences can be broadcasted by advanced protocols, and can be used to evaluate a router buffer with regard to the data load (like differentiating packages related to audio streaming or video frames). Studies on how to optimally (or at least in a fair manner) mirror the user preferences in data dropping decisions are subject of further exploration.

5. Conclusion

We have presented an approach to complete user answers for a questionnaire from partial user-supplied information. In this approach, from a corpus of existing user's questionnaire data, prototypes were generated. These prototypes represent typical answer patterns for all users, and allow to assign a particular prototype to each user. For assigning these prototypes to new users as well, the information of the new user can be given either by user attributes, or by only partially providing the same information as existing users. The feasibility of a user attribute-based approach could not be shown, as the corresponding classification failed to provide acceptable error rates. On the other hand, the partial-answer based approach gave good results, and appears to be feasible in practice. The trade-off between the number of clusters and prototype selection accuracy has been considered as well, indicating an example relation of prototype selection accuracy of about 70%, while only providing 30% of the data for new users and using 8 prototypes.

In future work, the trade-off between the number of clusters and accuracy will be further studied. In addition, in this work, the selection of the partial answers has been performed randomly. The existence of an optimized selection scheme might be expected. This issue, and also more specific ways of employing the yielded information in network routing and control, will be topics of future work.

References

[1] G. Adomavicius, A. Tuzhilin, Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions, IEEE Transactions on Knowledge and Data Engineering 17 (6) (2005) 734–749.

[2] G. Apostolopoulos, R. Guérin, S. Kamat, S.K. Tripathi, Quality of service based routing: a performance perspective, SIGCOMM Computer Communication Review 28 (4) (1998) 17–28.

[3] G. Armitage, Quality of Service in IP Networks: Foundations for a Multi-service Internet, Macmillan Publishing Co., Inc., Indianapolis, IN, USA, 2000.

[4] M. Boguna, D. Krioukov, K. Claffy, Navigability of complex networks, Nature Physics 5 (2009) 74–80.

[5] R.O. Duda, P.E. Hart, D.G. Stork, Pattern Classification, Wiley-Interscience Publication, 2000.

[6] P. Ferguson, G. Huston, Quality of Service: Delivering QoS on the Internet and in Corporate Networks, John Wiley & Sons, Inc., New York, NY, USA, 1998.

[7] G. Fischer, User modeling in human–computer interaction, User Modeling and User-Adapted Interaction 11 (1) (2001) 65–86.

[8] J. Gozdecki, A. Jajszczyk, R. Stankiewicz, Quality of service terminology in IP networks, IEEE Communication Magazine 41 (3) (2003) 153–159.

[9] J.A. Hartigan, M.A. Wong, Algorithm AS 136: a *K*-means clustering algorithm, Applied Statistics 28 (1) (1979) 100–108.

[10] R. Hecht-Nielsen, Confabulation Theory: The Mechanism of Thought, Springer-Verlag, 2007.

[11] A. Heß, N. Kushmerick, Learning to attach semantic metadata to web services, in: The SemanticWeb – ISWC 2003, 2003, pp. 258–273.

[12] A. Kobsa, User modeling and user-adapted interaction, in: CHI '94: Conference Companion on Human Factors in Computing Systems, ACM, New York, NY, USA, 1994, pp. 415–416.

[13] M. Koeppen, K. Yoshida, User modeling by confabulation theory, in: Proceedings of the 2008 IEEE Conference on Soft Computing in Industrial Applications (SMCia08), Muroran Institute of Technology, 2008, pp. 55–59.

[14] C. Lee, J. Lehoczky, R.R. Rajkumar, D. Siewiorek, On quality of service optimization with discrete QoS options, in: Real-Time and Embedded Technology and Applications Symposium, IEEE, 1999, p. 276.

[15] D.A. Menasce, QoS issues in web services, IEEE Internet Computing 6 (2002) 72–75.

[16] R. Mineura, M. Koeppen, K. Yoshida, Questionaire analysis using self-organizing map, in: The 9th Annual Meeting of Japan Society for Fuzzy Theory and Intelligent Informatics Kyushu Chapter, 2007, pp. 35–38.

[17] A.S. Tanenbaum, Computer Networks, second ed., Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1988.

[18] D. Towsley, V. Firoiu, J. yves Le Boudec, Z.-L. Zhang, Theories and models for internet quality of service, in: Proceedings of the IEEE, special issue in Internet Technology, vol. 90, 2002, pp. 1565–1591.

[19] Z. Wang, J. Crowcroft, Quality of service routing for supporting multimedia applications, IEEE Journal on Selected Areas in Communications 14 (1996) 1228–1234.

[20] I.H. Witten, E. Frank, Data mining: practical machine learning tools and techniques, in: Morgan Kaufmann Series in Data Management Systems, second ed., Morgan Kaufmann, June 2005.

[21] X. Xiao, L.M. Ni, Internet QoS: a big picture, IEEE Network 13 (1999) 8–18.

[22] K. Yoshida, Contents management system based on kansei information processing: Investigation for multimedia contents delivery service, in: Proceedings 2004 Conference on Intelligent Systems Design and Applications (ISDA04), Budapest, Hungary, 2004.