A Model for Short-Term Content Adaptation

Marco Benini
Centro di Ricerca Informatica
Interattiva
Dipartimento di Informatica e
Comunicazione
Università degli Studi
dell'Insubria
via Mazzini 5, Varese, Italy
marco.benini@uninsubria.it

Alberto Trombetta
Dipartimento di Informatica e
Comunicazione
Università degli Studi
dell'Insubria
via Mazzini 5, Varese, Italy
alberto.trombetta@uninsubria.it

Michela Acquaviva
Centro di Ricerca Informatica
Interattiva
Dipartimento di Informatica e
Comunicazione
Università degli Studi
dell'Insubria
via Mazzini 5, Varese, Italy
michela.acquaviva@uninsubria.it

ABSTRACT

This paper proposes a model for short-term content adaptation whose aim is to satisfy the contingent needs of users by adjusting the information a web-application provides on the basis of a short-term user profile. The mathematical model results in the design of an adaptive filter that profiles users by observing their queries to the application and that adjusts the answers of the application according to the inferred user needs. Also, the mathematical model ensures the correctness of the filter, that is, the filter is guaranteed to exhibit a coherent short-term adaptive behaviour.

Categories and Subject Descriptors

H.1.2 [**User/Machine Systems**]: Human factors; H.3.3 [**Information Search and Retrieval**]: Information filtering; H.5.4 [**Hypertext/Hypermedia**]: Navigation, User issues

General Terms

Algorithms, Human factors

Keywords

User modelling, Information filtering

1. INTRODUCTION

Among the many possibilities offered by the widespread use of the web, E-learning, has proved to be more difficult than initially thought. One central difficulty lies in the lack of interaction and synchronisation between learner and teacher. The solution offered by the most successful E-learning initiatives has resulted in a highly improved way of presenting contents, such as learning objects' standards. Despite this, the problem still remains.

As a partial explication, one can point out that little or no effort has been spent in order to overcome the lack of personalisation of the content, in terms of learner needs. Aim of this work is to present a framework to model the interaction between a learner and an E-learning service, and

to model the adaptive behaviour the E-learning service has to offer, in order to better satisfy the specific learner needs.

The adaptive agent is a filter that intercepts user queries to the learning objects' repository and uses them to define a corresponding user profile. Then, it adapts the repository's answer according to the adaptive strategy. The profile is a model of a user, and is devoted to code his interests.

In this work, we focus on the short-term behaviour, being it the real novelty in our adaptive approach. In fact, the long-term adaptive model is a natural extension of the short-term model, and is comparable with many existing approaches. Under the hypothesis that the sequence of queries represents an attempt to satisfy an interest of the user, we expect that every query in the sequence refines the interest, until a satisfying result has been found. Hence, the transformation process implemented by the adaptive filter should result in a sequence of answers that are closer and closer to the satisfaction of the user interest, and such sequence should converge with the same speed as the sequence of queries. Therefore, the adaptive filter must be able to segment the flow of queries from a user in sequences, each converging to the full expression of a single interest, until such interest gets satisfied. In other words, it should sense the changes of interest in the flow of queries issued by the user. The filter provides families of transformations, such as ordering the elements of a multiple answer according to their closeness to the user interests. Such transformations are parametrised with the profile of the corresponding user. The property the adaptive model is required to satisfy is informally stated as *to a small change in the profile, there corresponds a small change in the transformation*. Hence the need for a mathematical study of the adaptive filter.

2. A SHORT-TERM ADAPTIVE MODEL

The adaptive filter receives a query from the user, then it analyses the query with the *analyser* algorithm and it forwards the query to the repository of the application, i.e., to the part of the application deputed to answer queries, typically a database; finally, the filter adapts the answer from the repository by means of the *transformer* algorithm.

The goal of the adaptive strategy performed by the filter is to exhibit two behaviours: *convergence* and *detection of changes of interest*. The former means that if the sequence of queries approximates the user interest, then the series of transformed answers must be closer and closer to the sat-

isfaction of that interest. The latter behaviour, *detection of change of interest*, captures the situation where the user stops a sequence of queries and starts a new one, focused on a different interest.

In this respect, the user is modelled as a point in the *profile space*, denoted as $\mathcal{U}_{\text{profile}}$, the cube with edge $[-1, 1]$ in the n-th dimensional Euclidean space. Every coordinate of this space is uniquely associated with an attribute and its value denotes how strong is the user interest in the attribute.

Being a subspace of the Euclidean space, the profile space is equipped with a norm $\| \ldots \|$ that measures the *length* of a profile $(p_1, \ldots, p_n)$. Obviously, the associated distance function is $d(a, b) = \|a - b\|$, as usual, see, e.g., [1]. The norm is defined as $\|(p_1, \ldots, p_n)\| = \left(\sum_{i=1}^{n} p_i^2 \right)^{1/2}$.

Moreover, the zero vector $\vec{0}$ represents the *neutral profile*, modelling the user deprived from his interests.

The important point in our model is that a query can be interpreted as a profile. In fact, we assume that a query asks for metadata with a specified degree of confidence. Thus a profile can be immediately constructed: if a query asks for resources marked with the metadata a with a degree of confidence d, the corresponding profile will have the value d in the coordinate a; if the query does not specify any value for a coordinate a, the profile will have the value 0 in a.

Therefore, given a sequence of queries $q_1, \ldots, q_n$, it is possible to associate to every query a corresponding profile, obtaining a sequence of *query profiles* $\hat{q_1}, \ldots, \hat{q_n}$. Thus, given an initial user profile p_0, the sequence of queries $q_1, \ldots, q_n$ generates the series of user profiles $p_0, \ldots, p_{n+1}$ with the rule that $p_{i+1} = \alpha p_i + \beta q_{i+1}$, where $\alpha + \beta = 1$ and $\alpha, \beta > 0$. Hence, the new user profile is a *weighted mean* of the two old user profile and the query profile.

Consequently, the rule that identifies a change of interest is a lack of convergence in the sequence of profiles. A way to locally measure the convergence of the generated profile sequence is to observe if $\|p_{i+1} - p_i\| > \delta/f(i)$, where δ is a constant threshold value and f is a monotone increasing function. For example, if $f(i) = i$ then a sequence not violating the constraint converges with linear speed. In general, f measures the convergence speed of generated profiles.

Therefore, the analyser algorithm converts the input query q to a profile $\hat{q}$, then it calculates the new profile n as the weighted mean of $\hat{q}$ and p_i, the current user profile. If the distance between n and p_i is too large, i.e., greater than $\delta/f(i)$, then the analyser restarts a new query sequence, otherwise p_{i+1} becomes equal to n.

On the other hand, the transformer algorithm is designed according to the principle that a small variation in the user profile induces a small variation in the applied transformation. Thus, we assume that the filter is equipped with a set of transformations, and one of them is applied to the result according to the query and the user profile.

The transformation applied to the result depends on the query; to capture this dependency, the set Q of possible queries is partitioned by an equivalence relation ρ. Hence, the transformation space is the disjoint union of transformations: $\Theta = \bigsqcup_{x \in Q/\rho} \{\theta_x(p) \mid p \in \mathcal{U}_{\text{profile}}\}$. The functions $\theta_x : \mathcal{U}_{\text{profile}} \to (\mathcal{U}_{\text{answer}} \to \mathcal{U}_{\text{answer}})$ are the transformations. They act by generating a concrete function from a user profile, that, when applied to an answer, produces the transformed result. The result is a point in the space $\mathcal{U}_{\text{answer}}$, the set of possible answers, usually identified with the set of valid XML documents. The link between the transformation space and the profile space is given by a constraint that ensures the preservation of convergence: let $x \in Q/\rho$ be the selector of a transformation, let $a \in \mathcal{U}_{\text{answer}}$ be an answer from the repository, then we impose that, for every $pa, pb \in \mathcal{U}_{\text{profile}}$,

$$d_x(\theta_x(pa)(a), \theta_x(pb)(a)) \leq K d(pa, pb) \ , \qquad (1)$$

where d is the distance function in the profile space, d_x is the distance function in the answer space associated with the θ_x transformation, and K is a (small) constant, possibly depending on a and x.

Henceforth, a transformation θ_x is admissible iff is possible to associate a distance function d_x satisfying the constraint (1). In fact, if the distance function d_x is chosen such that the metric space $\mathcal{U}_{\text{answer}}$ is a Banach space, where every Cauchy sequence converges [1], then the constraint (1) is a direct formalisation of the sentence "a small variation in the profile space induces a small variation in the transformation", since the constraint ensures that there is an homeomorphism between the profile space and a suitable subspace of the transformation space.

As an example of admissible transformation, we consider the transformation that orders a list according to the proximity of its elements to the user interests. Precisely, an ordering transformation operates by sorting the list of elements $(r_1, \ldots, r_n)$ according to a sorting key that is a continuous function $f : \mathcal{U}_{\text{profile}} \times \mathcal{U}_{\text{profile}} \to \mathbb{R}$, where $\mathbb{R}$ is the set of real numbers. To every element r_i of the list $(r_1, \ldots, r_n)$ is associated a point $\hat{r_i}$ in the profile space, in a similar way as already described for converting queries into profiles. Since the function f is applied to the pair $(\hat{r_i}, p)$, p being the user profile, the list is sorted according to a value depending on f, p and the examined element.

Moreover, the ordering transformation satisfies the constraint (1). In fact, posing $d_x(a, b)$ as the minimal number of element swappings needed to transform the list a into the list b, it follows that d_x is, indeed, a distance function. Since f is continuous, but sorting is a discrete function, a sufficiently small value ϵ of $d(pa, pb)$ implies that $\theta_x(pa)(a) = \theta_x(pb)(a)$, thus it follows that $d_x(\theta_x(pa)(a), \theta_x(pb)(a)) \leq n/\epsilon$ because no more than n swaps are needed to transform a list of length n into any of its permutations. Thus, posing $K = n/\epsilon$, the constraint (1) immediately follows.

3. CONCLUSIONS

The contribution of this work is a mathematical model for short-term content adaptation. In contrast with most existing approaches oriented to the long-term observation of users, we have developed a model that is based on rough observations for a short period of time. Although this model is not precise as the long-term counterparts, it is powerful enough to capture the basic user interests and it provides a mathematical frame to guarantee the correctness of adaptive manipulations on content.

4. ACKNOWLEDGEMENTS

5. REFERENCES

[1] K. Jänich. *Topologie*. Springer Verlag, 1994.