

Research Article

On a Modeling of Online User Behavior Using Function Representation

Pavel Pesout and Ondrej Matustik

Faculty of Informatics and Statistics, University of Economics, 130 67 Prague, Czech Republic

Correspondence should be addressed to Ondrej Matustik, ondrej.matustik@vse.cz

Received 30 December 2011; Revised 14 June 2012; Accepted 16 June 2012

Academic Editor: Gerhard-Wilhelm Weber

Copyright © 2012 P. Pesout and O. Matustik. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Understanding the online user system requirements has become very crucial for online services providers. The existence of many users and services leads to different users' needs. The objective of this presented piece of work is to explore the algorithms of how to optimize providers supply with proposing a new way to represent user requirements as continuous functions depending on time. We address the problems of the prediction the of system requirements and reducing model complexity by creating the typical user behavior profiles.

1. Introduction

Ubiquitous computing represents a new generation of interaction with computers and is a promising way for users to obtain the needed services, as well as for the distribution companies to distribute their applications with lower distribution costs and attract customers for longer time period. It has emerged as a natural evolution step in computer sciences as the computer-based appliances are becoming smaller, more mobile, and more interconnected than ever before.

The aim of this paper is not to discuss the future potential of ubiquitous computing (as we believe that there are already enough articles concerning it) or to describe the general principles of computing and software engineering innovation adherent to ubiquitous computing, but rather to propose a new approach to modelling of ubiquitous computing online user requirements (further UCUR) and behavior, compute system level agreement with the UCUR provider, and reach the optimal infrastructure by user allocation among particular sources in order to fulfil the user's needs.

The ubiquitous computing service has not been very clearly defined yet. We will therefore assume, for the purpose of our paper, that ubiquitous computing service is every

regular service fulfilling ubiquitous computing requirements—that is, the service is accessible everywhere, fully integrated into everyday objects and activities and is not connected with any single type of hardware.

There are apparently some differences in the intensity of different service usage. For a company which provides more than one service, it seems to be only rational to adjust the level of services provided during the daily hours in order to optimize the usage of available resources and keep the level of services according to the client needs. We can assume that different users will require a different kind of services in different time, for example,

- (i) marketing user—during work time he/she uses mainly databases, storage, marketplace and email services, in his/her free time mainly web services;
- (ii) accountant—during work time uses mainly billing and accounting services, partly also infrastructure and databases; at the same time, it must be noted that the usage of services differs during the month, for example, the use of accounting and billing services intensifies during thanks to financial statements preparation, the use of data sharing increases during the financial audit, and so forth;
- (iii) school child—will use mainly web services, e-mail, and data sharing, however, in different time period than the full-time employee.

The examples shown above illustrate the differences in the intensity of service use.

In this piece of work we assume that our model company provides several different services and serves different types of users as mentioned above. This company operating model can in an optimal mode bring significant savings from the scale of the services provided; nevertheless, on the other hand it can cause some significant issues with the set-up and managing the system.

The service provider has to manage and monitor its services level provided at any time for each particular service. Based on our research, one possible way is of reaching the optimal level of the service management and monitoring lies in computing the system development and predicting its changes by the modeling of typical user behavior. Our approach is based on the possibility to model user needs as functions and then to use these typical user needs as a representative of one of the groups created by using function data clustering techniques.

The further parts of this paper are organized as follows. In Section 2, the current studies in the field of ubiquitous computing and curve clustering techniques are briefly reviewed. In Section 3, the user behavior and the attributes that influence it are described. The methods of functional data clustering used in order to create typical user behavior requirements are described in Section 4. In Section 5, the overall system requirements are computed and system changes are discussed. The algorithms of system source's allocation are proposed in Section 6. Finally, a conclusion is summarized in Section 7.

2. Related Work

Our work is not fully concentrating on the theory of the ubiquitous computing, although it influences our research deeply. The basic ideas of ubiquitous computing are very well summarized in the work by Greenfield [1]. The possibilities of future development in this field are named by many authors. Fano and Gershman [2] deal with some of them especially in the field of medical services and mobile wallet, and Yu with Guo [3] study the role of ubiquitous computing in retail banking.

We concentrate mainly on the ubiquitous computing online user requirements. Some authors even use the term *ubiquitous cloud computing* when they are writing about ubiquitous computing services, so the term cloud computing is in many aspects very close to ubiquitous computing itself. The ideas of cloud computing and its future role are very well described by Carr [4], who is mainly known for his comparison of IT systems to standard commodity rather than to a competition advantage. According to him, the switch to cloud computing services (and also to ubiquitous computing services) will be probably very similar to the switch from single electricity generators to the electricity grid.

In many science sources we find many classifications of the cloud computing services; very well known is the classification of cloud computing services by Cearley and Smith [5].

- (i) Infrastructure as a Service (IaaS)—basically raw compute and storage services—this option provides only an infrastructure without any software, so the modeling of the requirements is not particularly difficult.
- (ii) Platform as a Service (PaaS)—higher-level of development environments which abstract the underlying technology and provide for scalability and rapid application development.
- (iii) Software as a Service (SaaS)—classical online software provided as a service with minimal requirements on installation on user's computer and almost all data are stored on the provider's side.

For the purpose of this paper, we introduce our UCUR model under the category SaaS; however, we strongly believe that our model is also suitable for other cloud computing categories (even though they are not as complex as SaaS). Another view on ubiquitous computing services can be found in the work of Kim et al. [6], where the authors deal with the idea of Offload Socket Processing—these changes in the socket processing can be viewed as another method of resources optimization.

It is obvious that the UCUR model encompass diverse types of services. Weinhardt et al. [7] distinguish the following types of services: Infrastructure, Storage, Database, Business Process Management, Marketplace, Billing, Accounting, Email, Data sharing, Data processing, and Web services for Software as a Service approach (as a part of cloud computing services), and we can also apply this categorization to the ubiquitous computing services. There are of course some other services, like webgames, location-based services, and so forth, but they are generally included in the Web services category.

In our research we concentrate on searching for typical user behavior; hence, we need define and classify the clustering techniques. Clustering is a process of grouping data of similar character into the same class or same cluster. However, it is important to note that in case the data are measured as a function of a dependent variable such as time, which applies to our case (user requirements), the most frequently used clustering algorithms, such as hierarchical and partition-based ones, may not pattern each of the individual shapes properly. That is why we have to choose more sophisticated and special methods of functional data clustering that are able to attend to the whole space among the measurements and are not limited to the obtained measurements set.

Recently, density-based clustering methods using the Maximum Likelihood Estimation (MLE) have been mostly developed to recognize most homogenous partitioning of functional data. There are two types of the density-based methods which differ in their approach to the cluster memberships. Firstly, these memberships may be assumed to be some of the model parameters. These methods are called the Maximum Likelihood Approach

(MLA) methods. They are thoroughly described by Fraley [8] and Banfield and Raftery [9]. The classification is a twofold one.

- (i) The likelihood is analytical or approximately maximized over the jointed parameters.
- (ii) By the use of estimations the likelihood function criterion is maximized over the cluster memberships.

Secondly, it can be assumed that the cluster membership is a random variable, and, hence, the mixture models will be used. Due to the height and the criticality of computational cost of finding the global minimum, only the local one is looked for through the application of the Expectation-Maximization (EM) algorithm described by Dempster et al. [10]. The classification is processed in the following way.

- (i) The cluster memberships are iteratively estimated.
- (ii) The jointed model parameters are estimated using the membership probabilities.

Nevertheless, most of these methods are almost unusable for our model as their efficiency decreases if a considerable variability exists within each subpopulation or group, which can be expected when dealing with user requirements.

It is very important to keep the possibility of an individual to differ partly from one or more characteristics of his or her group yet still exhibit the underlying behavior that distinguishes this group from the rest. In fact, there are only two suitable basic methods which solve the problem of atypical data sets:

- (i) untraditional including of estimated regression coefficients into the K -means algorithm innovatively proposed by Tarpey [12],
- (ii) the random effects regression mixtures with a hierarchical model with a mixture on parameters at the top level and an individual-specific regression model at the bottom level which were studied by Gaffney and Smyth [11].

3. User Behavior

The form of our solution to the allocation problem is largely influenced by several issues which we are confronted within the process of the UCUR model establishment and management. Firstly, different ubiquitous computing users have different requirements for provided services, and their demands are placed at different time during the day. Secondly, we have to face the logical problem of task's backlog with large number of users. A purely individual approach to each of them cannot be assured. Therefore, we propose to modify the tasks by the use of modelling a much lower number of typical behavior profiles.

Let now N be the number of ubiquitous computing users. For each N we assume there is a sufficient history of their requirements' measures. These measures have to be assigned to several monitored attributes in the solution. Basic attributes characterizing the user needs are at least five:

- (i) type of requested service (web, data based, accounting, billing, etc.),
- (ii) requested memory for the service,
- (iii) requested computing power (both CPU and graphics card),

- (iv) hard disk space (for the operations and for saving of the results as well),
- (v) claims on the line capacity for the connection between the service provider and its users.

However, generally it is possible to include even more attributes, for example, the speed of response. This can increase the model complexity.

Let H be the number of attributes included in our model. In our analysis we have identified some underlying premises.

- (i) The measures have to stem from the aggregations of requirements from previous time frames so that the period between two data points is included therein.
- (ii) Number of measures has to be relatively large with only small time differences as the user access to services may change any time.
- (iii) Due to the restrictions included in the problem we assume the measures in the form of averages from a longer period of time (e.g., a month) rather than from a shorter period of time (e.g., one day).
- (iv) Due to different nature and needs of different attributes we consider various frequencies of measures.

Monitored functions are determined by many variables; some are more critical (e.g., connection time, bandwidth, user habits, day of week) than others (speed of computer control, etc.). In our model, we assume that user access to services within one time zone follows the biological needs and work rhythm.

Given the definition of measures mentioned above let be for the i th ubiquitous computing user, $i = 1, \dots, N$, the h th attribute, $h = 1, \dots, H$, and its m^h -length data points $\mathbf{x}^h = (x^h(1), \dots, x^h(m^h))^T$ represented by the vector $\mathbf{y}_i^h = (y_i^h(1), \dots, y_i^h(m^h))^T$.

However, these vectors are not sufficient if we want to understand the user requirements thoroughly by all means. In our opinion, it is more useful to model user behaviour as a continuous function depending on time which reflects its actual demand (behaviour). Taking into account this consideration we see \mathbf{y}_i^h as a vector of some function data points' measures.

To define a profile of typical behavior we have to deal with the clustering of tasks which we can build on the averages of overall user requirements classified into one cluster. Our aim is to find a partitioning into some certain number of groups so that objects in the same cluster have high similarity when considering their shapes and, at the same time, objects in the different clusters have lower similarity.

4. Typical Users' Profiles

To be able to define a typical behaviour profile we need to focus on the clustering techniques. As we have mentioned in Section 2, we may use both—random effects regression mixtures with a hierarchical model and plugging estimated regression—coefficients into the K -means algorithm according to functional data character.

The basic advantages of the K -means algorithm, which is in fact a specific case of the EM approach, are that it is a computationally less demanding solution and does not neglect naturally periodic character of measures. Moreover, there are also other reasons why we should consider the use of it.

- (i) All prerequisites of the algorithm are satisfied—trajectories belonging to one attribute are measured in the same data points and have the same length (e.g., one day).
- (ii) Measurements are naturally periodic (days in a week)—we are looking for the solution that does not neglect this fact.

The K -means algorithm iterative relocates, see Hartigan [13] for more details, the objects \mathbf{y}_i^h into S_k^h clusters; let us suppose $k = 1, \dots, K^h$, by minimizing the within-cluster variance

$$\sum_{k=1}^{K^h} \sum_{i: \mathbf{y}_i \in S_k^h} d^2(\mathbf{y}_i^h, \mathbf{c}_k^h), \quad (4.1)$$

where d is the considered distance and \mathbf{c}_k^h represents the S_k^h class centroid. However, in our model we consider functional data clustering. Following the Tarpey's research, an individual regression model can be used to estimate the functional responses at finite number of time points

$$\mathbf{y}_i^h = \mathbf{X}^h \mathbf{r}_i^h + \boldsymbol{\varepsilon}_i^h, \quad (4.2)$$

where \mathbf{r}_i is a $p^h \times 1$ vector of regression coefficients, \mathbf{X}^h is a $m^h \times p^h$ design matrix determined by the choice of basic functions and evaluated at \mathbf{x}^h , and $\boldsymbol{\varepsilon}_i^h$ is a $m^h \times 1$ vector of random errors. The estimated regression coefficients can be gained by using least-squares

$${}^1\mathbf{r}_i^h = \left(\mathbf{X}^{hT} \mathbf{X}^h \right)^{-1} \mathbf{X}^{hT} \mathbf{y}_i^h. \quad (4.3)$$

Therefore, a natural way to cluster curves \mathbf{y}_i^h is to apply the K -means algorithm to the elements of the matrix

$${}^1\mathbf{R}^h = \begin{pmatrix} {}^1r_{11}^h & \cdots & {}^1r_{1p^h}^h \\ \vdots & \ddots & \vdots \\ {}^1r_{N1}^h & \cdots & {}^1r_{Np^h}^h \end{pmatrix}. \quad (4.4)$$

According to the \mathbf{r}_i^h classification, we classify the trajectory \mathbf{y}_i^h and the requirements of the i th user in order to arrange the h th attribute into one of the S_k^h clusters.

It is obvious that the choice of the interpolation model is an important part of the model.

- (i) Clustering results can differ depending on how the curves are fit to the data.
- (ii) Clustering results can differ depending on how the data is weighted—for example, using the cubic B-spline and natural cubic spline, despite the fact that the fitted curves are identical, the clustering can yield different results.

Our primary question of interest is which interpolation method to use. With regards to the periodic data character, we propose to include Fourier interpolation.

Let us denote $d^h(x_j) = x^h(j)(\pi/6)$, $j = 1, \dots, m^h$, and assume even number of measurements (typically 12 daylight hours). For each object \mathbf{y}_i^h we are now looking for function

$$D_i^h(x) = \sum_{v=0}^{m^h/2-1} \left(a_{iv}^h \cos(m^h x) + b_{iv}^h \sin(m^h x) \right) + \frac{1}{2} \left(a_{im^h/2}^h \cos\left(\frac{m^h}{2} x\right) + b_{im^h/2}^h \sin\left(\frac{m^h}{2} x\right) \right), \quad (4.5)$$

where

$$a_{i0}^h = \frac{1}{m^h} \sum_{j=1}^{m^h} y_i^h(j),$$

$$a_{iv}^h = \frac{2}{m^h} \sum_{j=1}^{m^h} y_i^h(j) \cos(v d^h(x_j)), \quad v = 1, \dots, \frac{m^h}{2}, \quad (4.6)$$

$$b_{iv}^h = \frac{2}{m^h} \sum_{j=1}^{m^h} y_i^h(j) \sin(v d^h(x_j)), \quad v = 0, \dots, \frac{m^h}{2}.$$

We also project the data observed in $(1, m^h)$ into the interval $(\pi/6, 2\pi)$, and for each curve we obtain a set of $m^h + 2$ parameters (even though two of them are zero). This is an important advantage since the number of measures does not increase and does not complicate the computational cost.

However, the K -means algorithm has one important disadvantage—the number of clusters K^h has to be given in advance, which is not needed by the random effects regression mixtures model thanks to the ability to use the Bayesian Information Criterion (BIC) proposed by Schwarz [14]. What is more, the process is an iterative one which leads to the ability to choose the best result by setting the initial parameters and the rules of committing the individual steps. Finally, the trajectories do not have to be measured in the same data points and the measures belonging to different time series do not have to be of the same length (although, as we have mentioned above, these requests are not necessary parts of our model case).

Mixtures model uses interpolation (4.2) again. Let $\Theta^h = \{\boldsymbol{\theta}_1^h, \dots, \boldsymbol{\theta}_N^h\}$ be the parameters of this data level which allow us to model the individual trajectory behavior. We may assume the i th conditional distribution taking the form $f(\mathbf{y}_i^h | \mathbf{x}^h, \boldsymbol{\theta}_i^h)$ (for the h th attribute). At the top level of the hierarchy, there is another model that describes the distribution of the parameters \mathbf{r}_i^h of each individual. Let $\Phi^h = \{w_k^h, \boldsymbol{\varphi}_k^h\}$ be the parameters at this level where w_k^h is the probability that an observation belongs to the k th cluster and $\boldsymbol{\varphi}_k^h$ are the parameters of the distribution on \mathbf{r}_i^h according to the known group template $f_k(\mathbf{r}_i^h | \boldsymbol{\varphi}_k^h)$.

That is why the unconditional class membership for \mathbf{r}_i^h is a finite mixture model

$$f(\mathbf{r}_i^h | \Phi^h) = \sum_{k=1}^{K^h} w_k^h f_k(\mathbf{r}_i^h | \boldsymbol{\varphi}_k^h). \quad (4.7)$$

Since we know that

$$f(\Theta^h | \Phi^h) = \prod_{i=1}^N \sum_{k=1}^{K^h} w_k f_k(\mathbf{r}_i^h | \boldsymbol{\varphi}_k^h) \quad (4.8)$$

and the equality $f(\Theta^h, \Phi^h) = f(\Theta^h | \Phi^h) f(\Phi^h)$, we may use the maximum-a-posteriori (MAP)-based EM algorithm in order to produce consistent parameter estimates.

The EM algorithm consists of two steps. In the first E-step, the expected value of the complete-data MAP function is taken with respect to the posterior condition which is prior of the cluster memberships. We also evaluate the expected value \mathbf{r}_{ik}^h of \mathbf{r}_i^h given \mathbf{y}_i^h and $z_{ik}^h = 1$, where $z_{ik}^h = 1$ if \mathbf{y}_i^h is a member of the S_k^h cluster and $z_{ik}^h = 0$ otherwise, and we set the membership probabilities using in the previous M-step updated parameters to

$$w_{ik}^h = \frac{w_k^h f(\mathbf{y}_i^h | \bar{\Theta}^{h*}, \boldsymbol{\varphi}_k^{h*})}{\sum_{j=1}^{K^h} w_j^h f(\mathbf{y}_i^h | \bar{\Theta}^{h*}, \boldsymbol{\varphi}_j^{h*})}. \quad (4.9)$$

In the M-step this expectation is maximized over the parameters Φ^h and $\bar{\Theta}^h$. The complete-data MAP objective function is for the set Y^h of for all \mathbf{y}_i^h and Z^h of all z_{ik}^h given as

$$\ln \left[f(Y^h | \mathbf{x}^h, \Theta^h) f(\Theta^h, Z^h | \Phi^h) f(\Phi^h) \right]. \quad (4.10)$$

This yields the following form of the EM-based algorithm:

- (i) randomly initialized membership probabilities w_{ik}^h ;
- (ii) calculated estimates for $\{\Theta^h - R\}$ and Φ^h ;
- (iii) w_{ik}^h and \mathbf{r}_{ik}^h are made contemporary;
- (iv) loop to point 2 and its repetition until the expected value of the complete-data MAP function stabilizes.

Of course, the conditional distribution must be fitted with some real-world distribution and we recommend normal mixtures with Gaussian error term.

The main advantage of this method is the individual approach to each trajectory that is modelled by Eigen function and is allowed to be regulated by a parametric manner. It is also a very effective method in case considerably large innercluster variability exists in the model. The partitioning achieved by the use of the K -Means algorithm or random effects mixtures, may be represented by the vector \mathbf{z}_i^h , where $z_{ik}^h = 1$ if \mathbf{y}_i^h is a member of the S_k^h cluster and 0 otherwise.

Individually fitting functions from interpolation model (4.2), let us denote $D_i^h(x)$. By setting this, it is possible to compute the looked for typical requirements' profiles ${}^c D_k^h(x)$ as the averages of all objects that belong to the S_k^h cluster

$${}^c D_k^h(x) = \frac{\sum_{i=1}^N D_i^h(x) I[z_{ik}^h = 1]}{\sum_{i=1}^N I[z_{ik}^h = 1]}. \quad (4.11)$$

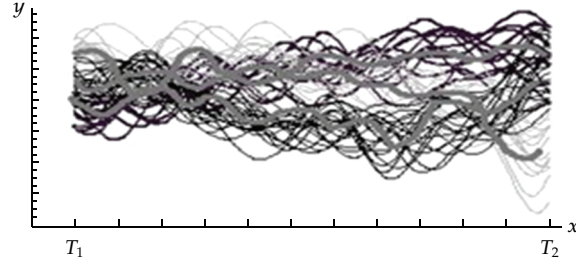


Figure 1: Experimental clustering results.

More transparently, the ${}^c D_k^h(x)$ functions should be transformed into nonaggregated forms denoted as ${}^{A,c} D_k^h(x)$. We have done some experiments in a general manner, and we can demonstrate some results of this kind of clustering technique in Figure 1. There are many users' requirements represented as particular curves and classified into three clusters. Bold curves display the profiles ${}^{A,c} D_k^h(x)$.

Note that all figures in this paper are only from experimental measurements and are not based on the real data.

To summarize the proposed process, the results are the following.

- (i) The requirements of the i th user to the particular h th attribute are modelled as the functions $D_i^h(x)$.
- (ii) Each function $D_i^h(x)$ is classified into one of the S_k^h clusters, which is identified by z_i^h values.
- (iii) Analytical view may be considered for only small number of the typical behavior profiles ${}^c D_k^h(x)$ computed thanks to partitioning of original objects.
- (iv) User individualities boil down to z_i^h values.

5. System Change and Overall Requirements

Modern UCUR is built on the premise that the needs of the enrolled users are constantly changing and evolving. However, this fact can be easily integrated into our profiles of typical behaviour.

- (i) If the i th user decides to withdraw from the system, then it will be sufficient just to remove (subtract) all the curves $D_i^h(x)$ from ${}^c D_k^h(x)$, where $z_{ik}^h = 1$. Recalculating the inclusion of other users is not necessary, assuming the high number of them.
- (ii) If we identify a new $(N + 1)$ th user, we will set its measurements $\mathbf{y}_{(N+1)}^h = (y_{(N+1)}^h(1), \dots, y_{(N+1)}^h(m^h))^T$ and will allocate them into one of the created clusters by using modified Fisher's canonical discriminant analysis, see its basics described by Fisher [15], in case that the K -means algorithm is used by clustering problem or discriminant functions (quadratic or linear discriminant scores) in case of random effects mixtures, see Ramsay and Silverman [16] for more details. Modification of the canonical discriminant analysis that we have specified consists in involvement of the estimation ${}^1 \mathbf{r}_{(N+1)}^h$ of the new requirement's regression coefficients using the model (4.2) of Fourier interpolation and least squares instead of original

measurements—it allows us to take into account the data functional character and to reduce the size of the task by detecting the most important data time points for classification.

- (iii) If we identify the change of needs of the i th user on the basis of the user complaints or explicit user behavior observation (or arbitrary correction), and we will remove all the curves $D_i^h(x)$ from ${}^cD_k^h(x)$, where $z_{ik}^h = 1$, we will update $\mathbf{y}_i^h = (y_i^h(1), \dots, y_i^h(m^h))^T$ and allocate it into one of the clusters in the same manner as if it were a new observation.

When there is the ability to reflect new or changing requirements by reallocating them into existing groups, we may focus on the detection of summary system requirements. Knowing the individual partitioning and profiles ${}^{A,c}D_k^h(x)$ allows us to give the summary h th attribute's requirements as the function

$$F^h(t) = \sum_{k=1}^{K^h} \sum_{i=1}^N {}^{A,c}D_k^h(t) I[z_{ik}^h = 1]. \quad (5.1)$$

The UCUR provider has now precise information about requirements on its system in any point in time, which helps to prevent the overwhelming system's sources. Nevertheless, to be able to make a decision about the approval or rejection of the new user application, it is necessary to know firstly how the sources are heavy and how the current requirements are allocated.

6. Requirements Allocation

To define our algorithm of requirements' allocation, we assume that the UCUR provider disposes with Ω sources (servers) of certain capacities on the h th attribute evaluated as ε_ω^h , $\omega = 1, \dots, \Omega$. The issue of requirements' allocation can be seen as a version of some clustering problem. We also propose the following process inspired by both the partitioned-based and agglomerative clustering algorithms.

Step 1. Choose randomly Ω user requirements identified with indexes $i(\omega)$ and allocate each of them to one source/cluster ${}^\Omega C_\omega$ that represents the ω th server with the fulfillment of the capacity condition ${}^{A,c}D_k^h(x) \leq \varepsilon_\omega^h$, where $z_{i(\omega)k}^h = 1$. Set the number of allocated requirements $s = \Omega$.

- (i) Denote by ${}^s\Psi_\omega^h(x)$ the actual charge of the ω th source, so now it is equal to ${}^{A,c}D_k^h(x)$, where $z_{i(\omega)k}^h = 1$.
- (ii) Denote by Ξ the set of all indexes which belongs to the allocated requirements, so now only all $i(\omega)$.

Step 2. For all sources and their charges we compute the cost/failure which is reached when adding particular number of profiles. We understand the cost as a state opposed to the optimal status. The most effective allocation is given by constituting the most constant requirement function as possible at any point in time for every attribute. Then we set for

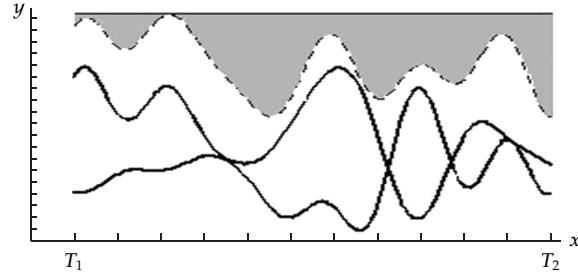


Figure 2: Possible profiles failure.

the h th attribute and weighted interval $[T_1, T_2]$ the $\Omega \times K^h$ matrix ${}^s\mathbf{G}^h$ of the elements ${}^s g_{\omega u}^h$ representing the possible cost:

$${}^s g_{\omega u}^h = (T_2 - T_1) \max_{t \in [T_1, T_2]} \left({}^s \Psi_{\omega}^h(t) + {}^{A,c} D_u^h(t) \right) - \int_{T_1}^{T_2} \left({}^s \Psi_{\omega}^h(t) + {}^{A,c} D_u^h(t) \right) dt. \quad (6.1)$$

The matrix determines the possible system costs/failures according to user's behavior. An example of this is illustrated in Figure 2. Bold curves represent typical profiles ${}^{A,c} D_v^h(x)$ and ${}^{A,c} D_u^h(x)$, a dashed curve shows the summation of these curves, the constant function is the maximum of its summation, and the filled area is the cost/failure $g_{vu}^h(t)$.

The least values of $g_{vu}^h(t)$ indicate the most optimal combinations of allocation. However, there are a lot of user requirements' classifications given by vectors \mathbf{z}_i^h , also the sum of all combinations must be computed. We have recalculated only the elements belonging to the sources which allocation was previously changed.

- (i) For the first requirement that has not been allocated yet, let it be the i th, i not in Ξ , and intend the cost of its allocation to ${}^s C_{\omega}$ with regards to all attributes

$$Q_{\omega i} = \sum_{h=1}^H e_h \sum_{\omega'=1}^{\Omega} \sum_{u=1}^{K^h} {}^s g_{\omega' u}^h I[z_{iu}^h = 1], \quad (6.2)$$

where e_h are the weights. The weights may be excided if all attributes are of the same importance, otherwise it is favourable to include them (their values depend on empiric UCUR provider's setting). Allocate the i th requirements to ${}^s C_{\gamma}$ if

$$Q_{\gamma i} = \min_{\omega=1, \dots, \Omega} Q_{\omega i}, \quad (6.3)$$

with the fulfillment of the capacity condition ${}^s \Psi_{\gamma}^h(x) + {}^{A,c} D_k^h(x) \leq \varepsilon_{\gamma}^h$, where $z_{ik}^h = 1$. Increase the number of allocated requirements $s = s + 1$. If the condition is not met, allocate it into the most optimal (according to $Q_{\omega i}$) source that meets that condition

(denote it again ${}^s C_\gamma$). In case such source does not exist, the requirement cannot be acquired.

- (ii) Update ${}^s \Psi_\omega^h(x)$ the actual charge of the ω th source, so ${}^s \Psi_\omega^h(x) = {}^{s-1} \Psi_\omega^h(x)$, where $\omega \neq \gamma$, and for the changed source ${}^s \Psi_\gamma^h(x) = {}^{s-1} \Psi_\gamma^h(x) + {}^{A,c} D_k^h(x)$, where $z_{ik}^h = 1$.
- (iii) Set $\Xi = \Xi U \{i\}$.

Step 3. Return to Step 2 until there are some nonallocated user requirements which have not been detected yet as unacceptable because of objectively limited source capacity.

7. Conclusion

In the age of ubiquitous computing it will definitely be needed to discover a new way to provide the services to customers, where both provider and customer benefit from simple and attractive revenue logic, which is no longer based on the application development investment and makes possible the achievement of the level of functionality, flexibility, and time to market required by users. On the other hand, the UCUR model ultimately changes the provider-customer relationship from one-to-one to one-to-many and to a typical-utility-based ecommerce relationship with a very crucial need to understand user's behavior and the effective algorithm of requirements' allocation. The model of the usage must be precise enough since the initial failure in the service delivery may result in the user's mistrust and the overcalling of service's hardware and software may result in high costs charges to the consumer or in the loss of the owner.

The primary motivation of this paper was to define the novel algorithm of user requirements' allocation between available sources. In order to achieve this aim we have introduced a new way to represent individual user needs as functional data measured in time for each of the attributes included. We have been focusing on finding the typical user behavior profiles by solving functional data clustering problem. Hence, we have integrated the linear regression into the K -means algorithm and dealt with the random effects regression mixtures with the EM algorithm and parameters at two levels. We have discussed the system development and changes of actual user demand, and we have demonstrated how to compute the overall system requirements. Based on executed research we are finally able to define the innovative algorithm that leads to the optimisation of UCUR sources' utilising and user allocation.

References

- [1] A. Greenfield, "Everyware—The dawning age of ubiquitous computing," Tech. Rep., New Riders, 2006.
- [2] A. Fano and A. Gershman, "The future of business services in the age of ubiquitous computing," *Communications of the ACM*, vol. 45, no. 12, pp. 83–87, 2002.
- [3] J. Yu and Ch. Guo, "An exploratory study of applying ubiquitous technology to retail banking," in *Proceedings of the Academies International Conference*, pp. 7–16, Academy of Banking Studies, Tunica, Miss, USA, 2008.
- [4] N. Carr, "The big switch' to cloud computing," <http://www.computerworlduk.com/>.
- [5] W. D. Cearley and M. D. Smith, "Cloud computing services: a model for categorizing and characterizing capabilities delivered from the cloud," in *Gartner Research ID: G00163913*, 2009.
- [6] S. Kim, S. Kim, K. Park, and Y. Chung, "Offloading socket processing for ubiquitous services," *Journal of Information Science and Engineering*, vol. 27, no. 1, pp. 19–33, 2011.

- [7] C. Weinhardt, A. Anandasivam, B. Blau et al., "Cloud computing—a classification, business models, and research directions," *Business & Information Systems Engineering*, vol. 1, no. 5, pp. 391–399, 2009.
- [8] C. Fraley, "Algorithms for model-based Gaussian hierarchical clustering," *SIAM Journal on Scientific Computing*, vol. 20, no. 1, pp. 270–281, 1998.
- [9] J. D. Banfield and A. E. Raftery, "Model-based Gaussian and non-Gaussian clustering," *Biometrics. Journal of the Biometric Society*, vol. 49, no. 3, pp. 803–821, 1993.
- [10] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society. Series B*, vol. 39, no. 1, pp. 1–38, 1977, With discussion.
- [11] S. Gaffney and P. Smyth, "Trajectory clustering with mixtures of regression models," in *Proceedings of the 5th ACM SIGKDD International Conference on Knowledge discovery-Data Mining*, pp. 63–72, ACM Press, New York, NY, USA, 1999.
- [12] T. Tarpey, "Linear transformations and the k -means clustering algorithm: applications to clustering curves," *The American Statistician*, vol. 61, no. 1, pp. 34–40, 2007.
- [13] J. A. Hartigan, *Clustering Algorithms*, John Wiley & Sons, New York, NY, USA, 1975.
- [14] G. Schwarz, "Estimating the dimension of a model," *The Annals of Statistics*, vol. 6, no. 2, pp. 461–464, 1978.
- [15] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of Eugenics*, vol. 7, pp. 179–188, 1936.
- [16] J. O. Ramsay and B. W. Silverman, *Functional Data Analysis*, Springer Series in Statistics, Springer, New York, NY, USA, 2nd edition, 2005.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

