Towards Extracting Realistic User Behavior Models

Reiner Jung
rju@informatik.uni-kiel.de
Kiel University, Germany

Marc Adolf
mad@informatik.uni-kiel.de
Kiel University, Germany

Christoph Dornieden
cdor@informatik.uni-kiel.de
Kiel University, Germany

Abstract

Workloads can be characterized by intensity and user behavior. Combining multiple intensities and behaviors can be used to create workload profiles to evaluate software design and support the prediction of system utilization. The central challenge for workload profiles is their fit to real workloads and in particular the match to specific behaviors. This is especially relevant for understanding and identifying specific user groups and support workload composition by operators.

In this paper, we address the identification of such realistic user behaviors utilizing domain specific attributes, evaluate the fitness of potential behavior clustering approaches, and discuss our setup to evaluate further clustering approaches.

1 Introduction

Service quality of software systems is influenced by workload intensity and the user behavior. Both factors play a vital role characterizing the system workload [8], which is relevant to understand past workloads and construct workload profiles to estimate future system utilization. For example, the resource consumption of browsing a catalog and purchasing items can be quite different. Therefore, it is necessary to be able to distinguish specific kinds of user behavior to characterize the workload sufficiently.

State of the art workload characterization approaches, such as WESSBAS [9], use a behavior mix, where different workload intensities are combined with specific user behavior models. These approaches collect user sessions and aggregate them to behavior models. WESSBAS estimates behavior models utilizing X-means clustering [3]. Such behavior models have two key shortcomings: (I) They reflect the observed behavior of the past, but might not represent specific user groups correctly harming predictability. For example, a *detergent shopper* might reappear frequently while a *sunscreen shopper* has a different seasonal profile. (II) X-means only yields acceptable results for small parameter vectors of at least ordinal values, but current behavior models are mapped to vectors. More model transitions imply more parameters, which harm clustering [3].

To construct realistic behavior models, user behaviors must contain domain specific attributes, e.g., to distinguish detergent from sunscreen shoppers. Therefore, we must extend classic behavior models with domain knowledge and identify an aggregation approach capable of handling large parameter sets.

In this paper, we compare the two clustering approaches X-means and Expectation-Maximization (EM) [2] in context of realistic user behaviors. We utilize, therefore, the iObserve analysis service [7], which can use different clustering approaches.

In the remaining paper, Section 2 discusses user behavior models. Section 3 introduces two clustering approaches for behavior models. Section 4 presents the concept of realistic user behaviors, and Section 5 discusses preliminary results regarding the clustering approaches. Finally, Section 7 summarizes our findings and discusses further research.

2 Behavior Models

Behavior models describe kinds of users and are aggregations of single user behaviors with similar behavior patterns. A single user behavior comprises all system invocations (entry level events) of a user during a session. It can be modeled as a path over visited pages or transformed into a behavior graph or a Markov-chain, which may contain loops for repetitive behavior. These paths or graphs are then grouped for similarity and merged into a behavior model.

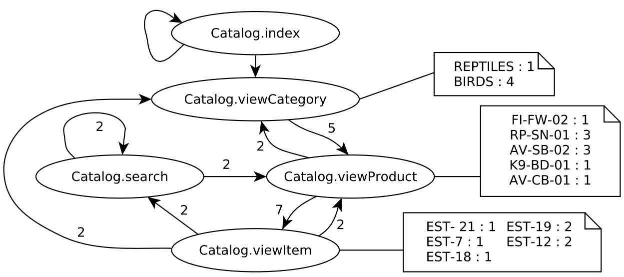

Figure 1: Behavior graph representing a shopper from our JPetStore [12] use case.

Figure 1 depicts an annotated behavior graph of a user interacting with the JPetStore [12], an example application resembling a shop system for pets. In this graph, nodes represent page visits and edges express the transitions between pages. The numbers at the edges indicate the amount of transitions between pages. In addition, we added domain specific information, like the viewed category and product, which can be used to support the behavior clustering.

3 Clustering Approaches

Clustering can be used to identify groups of data points which share similarities. In our context, we use clustering to identify user behavior models, like WESSBAS, which uses the X-means. Clustering is affected by density, distances, and distribution of data points. Depending on the clustering approach, the dimension of the data points can have a significant impact on the quality of the clusters. We employ two clustering methods provided by Weka [4]:

X-Means X-means builds on the K-means clustering algorithm [3], which consists of three steps [1]: (1) For every expected cluster (K), a center point, called centroid, is randomly chosen from the data points. (2) According to a chosen distance metric, each point x of the data set is assigned to the closest centroid. (3) The centroids are recomputed according to the center of mass of the points belonging to it. (2) and (3) are repeated until a convergence criterion is met.

In contrast to K-means, X-means searches over a range (e.g., 2 to 10) for a set of clusters, which provide the best fit. Therefore, X-means starts with computing K-means for the lower bound (e.g., K=2). Subsequently, each cluster is split into two using 2-means to try to improve the fit. Both steps are iterated while incrementing K until the upper bound is reached or an iteration is worse than the one before [3].

Expectation-Maximization EM is an iterative method consisting of two phases (E- and M-step) that are repeated until the convergence criteria is met and a final set of clusters is identified. Initially, a random set of cluster identifying data points are defined which are the initial parameters for EM. The *E-step* uses these parameters to compute the expected values of each data point. The *M-step* uses the E-step results to compute a new maximum likelihood for the data points regarding the parameters. This way, the new parameters for the next iteration are computed. These two steps are repeated until the convergence criterion is reached [2].

4 Realistic User Behaviors

We define realistic user behavior models as behavior models which reflect real groups of users in contrast to approximated groups, i.e., groups solely defined by their transitions, neglecting domain-specific data. For example, our detergent shoppers should form a separate group from those buying sunscreen. This is helpful to better understand seasonal behavioral changes and allow to create and modify workloads more realistically. This is relevant in scenarios, where workload characterizations can be modified to provide the system with knowledge of upcoming events, like a sunscreen shopper just before the holiday season.

A key ingredient for realistic user behaviors is domain-specific data, like the products or categories. With this additional data, user behavior can be classified in different groups. To be able to use such values in a clustering approach, a suitable metric must be defined, e.g., products of similar type must be closer together than products which are in another category.

5 Evaluation Setup

Our evaluation is based on JPetStore [12] instrumented with Kieker [5]. We modeled five realistic user behaviors [10], which utilize all functions of the JPetStore. The behaviors are tailored to share common behavior, but also include significant differences regarding pages, transitions, and request parameters, e.g., whether the person shops cats or fishes:

Account manager (AM) Changes contact information after login. Inspects one of the prior orders.

Browsing user (BU) Search products and only browses categories, products, and items.

Product lover (*L) Visits the CATS (CL) or FISH (FL) category and selects one product. Repeats 8 times and concludes shopping.

Single product buyer (S*) Goes to a category (REPTILES (SR) or CATS (SC)) and buys one item.

New customer (NC) Registers as a new customer, logs in, and buys a reptile.

We used these behavior models to create workloads with Selenium [13] and defined a set of seven ideal behavior models (IBM) for the analysis to detect. We executed JPetStore together with our workload and collected monitoring data. The data was then processed by the iObserve analysis [7] using different clustering algorithms provided by their respective filters.

The X-means setup is based on preliminary work, where we tested different configuration parameters for the algorithm [10]. We choose a configuration for X-means which provided the best fit to the JPetStore scenario. We set the range for the number of expected clusters to [6..12] and use the Manhattan Metric. For the EM clustering, we decided to go with the standard setting in Weka and are not setting any parameters, including the pre-estimated number of clusters. Since both algorithms start with randomly chosen values, the results may differ between each execution. Therefore, we execute each clustering five times to avoid results solely based on arbitrary starting values. In X-means, the resulting user behaviors are computed based on all behaviors in one cluster. They do not necessarily correspond to a real behavior. In contrast, the EM clustering only groups measured behaviors. In this evaluation, we simply took one representative behavior of each group. This can be improved, e.g., by creating a mean vector of every cluster.

At the end of each analysis run (EM and X-means), we compared the detected clustered behavior models with the prepared IBMs. First, we identified which detected behavior model matches best to an IBM. Second, we identified the distance of the matching model. In case a match cannot be identified, this counts as

Table 1: Comparison between the generated clusters and the IBMs (scores and content).

Scores	AM	BU	CL	FL	SC	SR	NC
EM	0.25	0	0	0	0.8	0.8	0.9
X-means	0.25	0	0	0	0.1	0.1	0.4
Content	AM	BU	CL	FL	SC	SR	NC
EM		+	+	+	(a)	(c)	(e)
X-means		+	+	+	(b)	(d)	(f)

a miss (score=0). The match between two models is computed in three steps: (1) we remove nodes which are not connected to the behavior graph, as they are created by mapping graphs to matrices and back. (2) we identify missing and additional nodes and edges, and compute ratios between these differences and the IBM. The lower the ratios, the better the fit of the detected behavior model. (3) we compare the request parameters in the behavior models. For example, the IBM defines that parameter CATS appears once, but the detected behavior model includes REPTILES, then the behaviors do not match.

6 Results

Both approaches could not detect all seven IBMs (EM=4 and X-means=5 clusters), but some detected models matched an IBM. Table 1 depicts the scores of the best matching behavior and the content match.

EM and X-means both detected the *account manager* behavior, but there where minor discrepancies between the aggregations and IBMs. As we did not record parameters for these pages, we could not compare the behaviors content wise. The *browsing user*, *cat lover*, and *fish lover* where detected correctly by both algorithms. The *single cat buyer*, however, could not be detected by EM, the closest match was the *single reptile buyer* behavior (a). X-means created a merged cluster of cat and reptile buyer, and the new customer, identifiably by 1/3 possibility for a cat and 2/3 for reptiles (b). Similarly, the *single reptile buyer* was identified by EM (c) and X-means closest match was the same as for the *single cat buyer* (d). Finally, the *new customer* detection failed, as the returned cluster deviated significantly from the IBM. Also the found graph better matches a single buyer or product lover than the new customer.

7 Conclusion

In this paper, we evaluated the detection quality of EM and X-means clustering for behavior models. We used behavior models enriched to include domain-specific knowledge in the clustering process.

Our current findings are that both clustering approaches are able to differentiate some behaviors based on parameter information, which is an improvement in comparison to the clustering without this information. However, they are unable to detect all behaviors correctly. A key issue of X-means is that larger vectors resulting in less precise clustering [3]. While the default configuration and the behavior model merge mechanism for EM might be amendable. You may find the data, notes, and artifacts in our replication package [11].

In future, we will evaluate further classification algorithms, including neural networks [6], and investigate whether seasonal factors support the clustering. Furthermore, we will perform a parameter study to identify the optimal configuration for each algorithm. Finally, we will test our hypothesis with other domains, e.g., a wiki and ticket system.

References

[1] J. B. MacQueen. “Some Methods for Classification and Analysis of MultiVariate Observations”. In: *5th Berkeley Symp. on Math. Statistics and Probability.* Vol. 1. UC Press, 1967.

[2] T. K. Moon. “The expectation-maximization algorithm”. In: *IEEE Signal Process. Mag.* 13.6 (Nov. 1996).

[3] D. Pelleg et al. “X-means: Extending K-means with Efficient Estimation of the Number of Clusters.” In: *ICML*. Vol. 1. 2000.

[4] M. Hall et al. “The WEKA Data Mining Software: An Update”. In: *SIGKDD Explor. Newsl.* 11.1 (Nov. 2009).

[5] A. v. Hoorn et al. “Kieker: A Framework for Application Performance Monitoring and Dynamic Software Analysis”. In: *Proceedings of ICPE 2012.* ACM, Apr. 2012.

[6] H. B. Demuth et al. *Neural Network Design.* USA: Martin Hagan, 2014.

[7] R. Heinrich et al. “Architectural Run-Time Models for Operator-in-the-Loop Adaptation of Cloud Applications”. In: *Proceedings of MESOCA*. IEEE Computer Society, Sept. 2015.

[8] M. C. Calzarossa et al. “Workload Characterization: A Survey Revisited”. In: *CSUR* 48.3 (Feb. 2016).

[9] C. Vögele et al. “WESSBAS: extraction of probabilistic workload specifications for load testing and performance prediction—a model-driven approach for session-based application systems”. In: *SoSyM* (Oct. 2016).

[10] C. Dornieden. “Knowledge-Driven User Behavior Model Extraction for iObserve”. Master thesis. Kiel University, June 2017.

[11] *Replication package.* `doi.org/10.5281/zenodo.883095`. 2017.

[12] *MyBatis JPetStore application.* `http://www.mybatis.org/spring/sample.html`. 2017.

[13] *Selenium browser automation.* `http://www.seleniumhq.org`. 2017.