Evaluation of the Employment of Machine Learning Approaches and Strategies for Service Recommendation

Jens Kirchner[1,2](✉), Andreas Heberle[1], and Welf Löwe[2]

[1] Karlsruhe University of Applied Sciences, Karlsruhe, Germany
{Jens.Kirchner,Andreas.Heberle}@hs-karlsruhe.de
[2] Linnaeus University, Växjö, Sweden
{Jens.Kirchner,Welf.Lowe}@lnu.se

Abstract. Service functionality can be provided by more than one service consumer. In order to choose the service with the highest benefit, a selection based on previously measured experiences by other consumers is beneficial. In this paper, we present the results of our evaluation of two machine learning approaches in combination with several learning strategies to predict the best service within this selection problem. The first approach focuses on the prediction of the best-performing service, while the second approach focuses on the prediction of service performances which can then be used for the determination of the best-performing service. We assessed both approaches w. r. t. the overall optimization achievement relative to the worst- and the best-performing service. Our evaluation is based on data measured on real Web services as well as on simulated data. The latter is needed for a more profound analysis of the strengths and weaknesses of each approach and learning strategy when it gets harder to distinguish the performance profile of the service candidates. The simulated data focuses on different aspects of a service performance profile. For the real-world measurement data, 97 % overall optimization achievement and over 82 % best service selection could be achieved within the evaluation.

1 Introduction

Service-Oriented Computing (SOC), Software as a Service (SaaS), Cloud Computing, and Mobile Computing indicate the development of the Internet into a market of services. With little to no knowledge about the service implementation or the system environment, service consumers can dynamically and ubiquitously consume service functionality. Besides the actual functionality, service consumers are interested in the service performance, which is expressed in its non-functional properties (NFPs) such as response time, availability, or monetary costs. In such a service market, the same service functionality may be provided by several competing service providers. Among these similar services, service consumers are interested in the service which fits best to their (NFP) preferences. In particular, consumers are interested in the actual experienced performance.

© IFIP International Federation for Information Processing 2015
S. Dustdar et al. (Eds.): ESOCC 2015, LNCS 9306, pp. 95–109, 2015.
DOI: 10.1007/978-3-319-24072-5_7

One of the major characteristics of a service market such as the Internet is perpetual change. Entering and leaving service providers as well as the complexity of service dependencies and environments make service selection and recommendation a challenge. Our service recommendation framework uses a collaborative knowledge base of consumption experiences of similar consumers in the past to predict the performance of a service in a certain consumer-based call context. This is used to recommend the best-fit service candidate to a consumer, considering his/her preferences [1,2]. Since the experienced performance at consumer side is influenced by a consumer's call context, e. g., calling time and location, the performance has to be predicted based on this context. Furthermore, performance is different for different consumers who value the NFPs of a service differently. For instance, some consumers are more interested in a fast response time and rather neglect higher monetary charges than others who want to have a service for free and rather experience higher response times. Therefore, service value is individual and it has to be determined individually whether a service is actually best-fit in a specific context. Considering these aspects, we analyzed two machine learning approaches which are based on classification and regression in combination with learning strategies for an optimal employment within service recommendation. Our practical assessment is based on real-world measurement data as well as profile-guided simulation data which addresses certain aspects of changes in the performance behavior of services for a more fine-grained analysis.

2 Recommendation Background and Framework

As written above, perpetual change is one of the major characteristics of the Internet as a service market. Among similar functional services, the selection of a service instance is based on one or more NFPs. NFPs have different scales of measurement with different optimization functions. For example, response time is a ratio scale with an optimization towards the minimum. The availability of a service at a specific time is nominal: a service is either available or not. In such a case, the optimization focus is to select a service instance which has the highest (maximum) probability of being available. When the selection of a service instance is based on more than one NFP, NFP data has to be normalized in order to be comparable and calculable. In such a case, not all NFPs are equally important, so their importance has to be weighted and taken into account [2]. In [1], we introduced our framework which optimizes service selection based on consumer experience, call context, and preferences (utility). Within this paper, the focus is set on the machine learning approaches which can be employed for service recommendation in general, and which can be implemented within our framework. Figure 1 depicts how the broker component in our framework works and where machine learning methods are employed. Constantly collecting the measurement data of service calls, the data is then pre-processed for the learning of each NFP of a service instance considering the contexts of the service calls from where the data derives. The NFPs/performance of each service instance are learned and constitute a background model. For each utility function and each

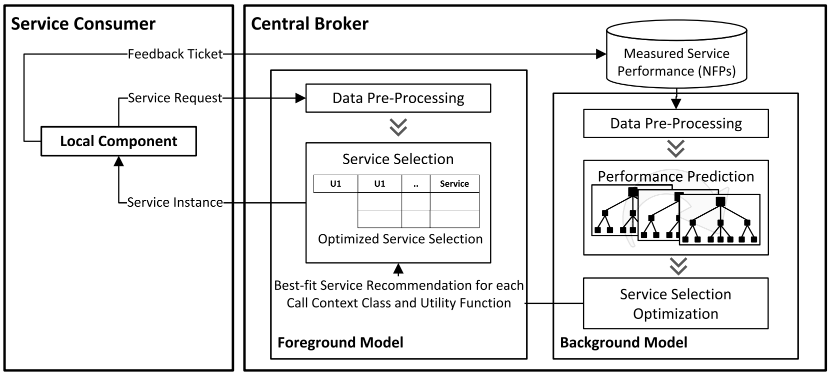


Fig. 1. Foreground and Background Model within our Framework [2]

context, this model contains the utility value of each service instance. Within each utility function and call context, the instance with the highest utility value is considered to be the best-fit service instance among functionally similar services. This best-fit service is saved in a foreground model to provide good performance within the time-critical recommendation process. It is updated with relearned background model information within intervals and on detected change. Once again, since preferences vary among all service consumers, the best-fit service instance is individual for each utility function.

3 Employment of Machine Learning Approaches

Initially introduced in [2,3], for the employment within our framework, we follow two machine learning approaches for service recommendation.

3.1 Regression

In machine learning, regression aims at the prediction of numerical values based on attribute values. Within our focus, regression can be used to predict each NFP value (e. g., the expected response time) based on call context values (e. g., calling time and weekday). The drawback is that learning has to be conducted for each NFP individually and the actual utility value for the best-fit determination has to be calculated. These higher efforts, however, have several benefits at the same time. For each NFP and call context combination, the NFP value has only to be predicted once, while the best-fit service can be calculated for each preference (utility function) individually. Furthermore, since each service's utility value is calculated (based on the expected NFP values) and considered, the ranking of second, third, etc. best-fit services can be used to achieve a higher overall performance gain. Note that utility optimization does not require a high accuracy of selecting the best-fit if the second best-fit achieves an almost as high utility gain. Also, underdog and quick starter strategies can also be implemented, since the performance data of services of calls of the past still remain.

3.2 Classification

Classification focuses on the determination of the affiliation to a certain class based on attribute values. In the recommendation scenario, consumers are ultimately interested in the selection of the best-fit service. So for a call context, the services could be classified into *best-fit* and *non-best-fit*. With this approach, the learning method focuses directly on the best-fit determination. For this, the training set has to be pre-processed: the champion has to be determined based on the measurement data. As a result, classification directly determines the best-fit service within a call context and utility function combination. The benefit using classification is to omit the calculation steps after prediction. Disadvantageously, however, the best-fit service has to be learned for each call context and each utility function; whereas having the NFPs predicted for a call context as an intermediate step, the best-fit service can be calculated for other utility functions without new learning. Furthermore, old service instances are automatically not further considered and, hence, sorted out. Disadvantageously, underdogs can never prove themselves since the approach is only focused on best-fit service recommendation and non-best-fit are neglected or not invoked at all. Also, there is no differentiation among non-best-fit services, which is important in a non-accurate prediction in order to still create a high utility gain.

4 Research Question

In [2,3], we showed that in general a classification- and regression-based approach can be employed for service recommendation. Additionally to these initial analyses, the present paper focuses on the research question regarding optimal learning strategies for the employment of machine learning approaches in service recommendation within a real-world scenario and focusing on certain performance behavior profiles. Within this, the following sub-questions address different aspects which need to be analyzed: What is the optimal size for the training dataset of a learning model in this context? How long is a trained model reliable for good service recommendation? Which learning strategy is better: incremental learning or sliding windows? Based on that, is drift detection implemented by a learning method better to cope with change? Is there a difference in the accuracy of service recommendation for different learning approaches when the services' performance profile becomes more and more similar?

5 Analysis of the Learning Approaches

For the conduction of the analysis, we developed a Java-based software that uses Weka[1] and MOA[2] as machine learning frameworks. Both were chosen because of their extensive collection of classical and state of the art methods and their

[1] http://www.cs.waikato.ac.nz/ml/weka/
[2] http://moa.cms.waikato.ac.nz/

Table 1. Evaluation Indicators

Overall Achievement	It indicates the optimization degree in per cent between the response time of the worst service towards the response time of the best service for each prediction case: $\left(1 - \frac{RT_{Prediction} - RT_{Best}}{RT_{Worst} - RT_{Best}}\right) \cdot 100$
Best Choice	It expresses the amount in per cent of the prediction of the actual best(-fit) service candidates.

high automation degree. Based on good initial results in regression [2], we chose the Fast Incremental Model Tree with Drift Detection (FIMT-DD) algorithm for regression. FIMT-DD focuses on time-changing data streams with explicit drift detection [4]. For classification, we chose DecisionStump[3], which showed the highest accuracy in initial experiments.

There are several aspects relevant for the evaluation of machine learning methods such as speed, accuracy, scalability, robustness, and interpretability [2,5,6]. Since service consumers are interested in the improvement of the performance they experience, we chose accuracy and performance as key indicators. As our focus is not set on the complexity of multi-target NFP optimizations, we reduce utility to the (arguably) most important NFP which is *response time*. The performance gain is therefore not based on a utility value, which is the weighted sum of different NPFs, but an *overall performance achievement* between the response time of the worst-performing service and the best-performing service. Second indicator is *best choice* to reflect the top-selection accuracy. The definitions of both indicators are listed in Table 1. Recall, service recommendation is an optimization problem. Hence, performance gain is more important than a high accuracy of selecting the best-performing service, since a slightly worse second best might still create a high performance gain which would not be reflected in best choice.

Regression is able to exploit continuous (date)time values. For classification, additional enhanced time attributes were added. Furthermore, in addition to the basic attributes *date*, *time*, and *response time*, we added further attributes to each measurement entry in order to provide statistically enhanced data with focus on natural periods. The additional attributes were added to all datasets. Note that these additional attributes (Table 2) only consider attribute values of previous records, since the current response time value of each record is part of the actual learning.

Due to different prerequisites of the two machine learning approaches, additional approach-specific pre-processing had to be conducted. Using regression, each NFP of each service instance has to be learned individually. Hence, the datasets had to be filtered into sub-datasets for each service. Classification is only interested in the best service at each moment in time. Therefore, the dataset had to be filtered so that only the records of the fastest services remained.

[3] http://weka.sourceforge.net/doc.dev/weka/classifiers/trees/DecisionStump.html

Table 2. Statistical Enhancement of Attributes

DayOfMonth	Extracted day of month from date
Hour	Extracted hour from time
Weekday	Determined nominal day of week from date *(class. only)*
Workingday	True is weelday is Monday to Friday *(class. only)*
RT_Xmin_AVG $X = \{61, 121, 181, 361, 721, 1441, 2881, 7201, 10081\}$	Response time mean of all records (chronologically) within the last 1, 2, 3, 6, 12 hours, and 1, 2, 5, 7 days
RT_X_AVG $X = \{40, 80, 160, 240\}$	Response time mean of the previous x records (chronologically; without consideration of any other attribute)
RT_X_AVG_Hour $X = \{4, 12, 20, 28\}$	Response time mean of the previous x records within the same hour value (1, 3, 5, 7 days of the same nominal hour)
RT_X_AVG_Weekday $X = \{4, 8, 16\}$	Resp. time mean of the prev. x records within the same weekday value (1, 2, 4 weeks of the same nom. day of w.)

Furthermore, the response time attribute is not directly used in classification, only indirectly for determining the best-fit service in the training set. Therefore, we removed the response time attribute for learning. After pre-processing, each dataset had to be divided into a training and a validation sub-set. Because of chronological aspects, the dataset could only be split into training and validation sets. N-fold cross validation could not be applied for that reason. In contrast to initial analyses, we conducted a sliding split point evaluation. For each analysis of an aspect, the split point between training set and validation set was iterated day by day. Depending on the period (and window sizes), it could result in a statistical mean of up to $\frac{n}{24m} - 1$ iterations (for the measurement input 170 iterations per scenario), for n data entries and m data records per hour (one record for each service).

5.1 Learning Strategies

In order to address the research questions related to the optimal learning strategies, we conducted the following scenarios. For both scenarios, we applied a prediction window of various sizes in order to determine the optimal training/prediction interval ratio for the updates of the foreground model (Figure 1).

Incremental Learning. This scenario continuously updates the learning model. Any strategies on changes and their impact on the model have to be dealt by the learning method such as drift detection.

Sliding Window Learning. This learning scenario applies a fixed window of previous measurements for the training of the learning model.

5.2 Measurement Data

The measurement data was gained from four real-world stock quote Web services [1]. The services are functionally similar, so they can be substituted.

The measurement period for this dataset was 185 days and contained 16,441 measurement entries. Each entry contained the date, time, the consumed service, and the measured response time. Within this period, each Web service was called on an hourly basis and its response time during this call was measured; hence, four data entries per hour. If a service was not available or timed out (30,000 ms), its entry was not added to the set.

5.3 Simulation Scenarios

The simulation of measurement data enabled to challenge machine learning approaches and to analyze their performance in certain scenarios. Within the measured real-world Web services, the statistical characteristics showed easily distinguishable performance profiles of the services. In order to compare the strengths and weaknesses of the machine learning approaches in more challenging scenarios, where the service profiles are harder to distinguish, we step by step approximated these profiles. For each profile, learning approach, and learning strategy combination, we conducted approximation iterations in 10 % steps until their profiles were fully identical (up to random noise).

Normal Distribution Profiled Data. As a baseline of every service profile, we assumed normally distributed response times of Web services around a mean value (with a certain standard deviation and variance). We created[4] normally distributed response time data for four services with a similar initial mean (vertical shift), standard deviation, and variance as the four measured Web services. This mean response time gets approximated step by step. Fully approximated, their statistical mean is identical.

Cyclic Spikes Up/Down. We generated two profiles with response times normally distributed around the same mean but with cyclic/periodic spikes which go in one profile up and in the other profile down. Spikes going up simulate services that have suddenly longer response times, while spikes going down simulate sudden response time improvements. For their creation, we used a saw tooth generator[5] in combination with an iceberg filter which are added to the basic normal distribution line. Again, all created services are similar. They are distinguished only in their horizontal shift. Fully approximated, their horizontal shift is identical.

Acyclic Spikes Up/Down. This profile has several acyclic spikes and different levels shifts in combination with an iceberg filter. Using several cyclic spikes in spikes generations with very long periods in combination with pulse train shifts[6] and the iceberg filter, a complete acyclic/aperiodic behavior could be simulated. Again, all services have the same mean response time and in a fully approximated case, their spikes are overlapping.

[4] Java implementation based on http://info.michael-simons.eu/2013/02/21/java-implementation-of-excels-statistical-functions-norminv/.

[5] Cf. Saw tooth generation using Fourier series http://mathworld.wolfram.com/FourierSeriesTriangleWave.html

[6] Fourier series expansion was used; cf. http://en.wikipedia.org/wiki/Pulse_wave

6 Results

The results presented in this section are mean values. For each input dataset, each evaluation scenario was conducted on a sliding iteration with sliding start, split, and end points in order to get statistically profound results.

6.1 Measurement Data

Based on the initially analyzed Web Services [3], we used an extended measurement dataset of over six months of four functionally substitutable real-world Web services. In contrast to the initial analysis, we conducted sliding iterations with sliding start, split, and end points. The purpose of these sliding iterations is to statistically equalize any start, split, or end point, so that the results are not influenced by any selected point. Within the evaluation of the two learning approaches, we used two general learning scenarios. The first scenario trains the learning model on an incremental basis, while the second uses a sliding window for training. For the window scenarios, we evaluated different window sizes.

Addressing the question whether increment or window learning is better, Table 3 shows the results of the incremental learning scenarios with prediction windows of 1 and 28 days. Within the overall optimization achievement of response time as well as the best choice indicator, there is not much deterioration of the predictions of the upcoming day to four weeks. However, the standard deviation of the results decreases for the wider prediction window; especially, in the case of the FIMT-DD, which we presume already due to its drift detection.

Analyzing optimal sliding training window sizes and the reliability within increasing prediction window sizes, we conducted sliding iterations with different window sizes for training and prediction. Table 4 reveals the statistical mean values for the analyzed training window sizes, while Table 5 for the analyzed prediction window sizes. Again, the analysis data revealed not much difference between the different window sizes in the overall achievement indicator, while there is a difference in the best choice indicator (Table 5). For the classification approach using DecisionStump, small prediction windows achieve better best choice indicator values than wider ones. Regression-based FIMT-DD remains more or less steady regardless of the window size. In direct comparison, the classification-based determination of the best-fit service is in general slightly better (in the case of a prediction window size of one day). In Figure 2(b), it seems that for the DecisionStump approach, a smaller prediction window results in a better best choice indicator, while for FIMT-DD a wider prediction window leads to a steadier best choice prediction (similar best choice mean, but a smaller standard deviation). Recall, the indicators introduced in Table 1 focus on different aspects. While the best choice indicator focuses on the accuracy in the overall prediction of the actual best-fit service (employing machine learning methods as well a calculation steps), the overall achievement indicator expresses the degree of optimization. Although it is desirable to always predict the best service candidate, for an optimization it is not necessary if the second-best creates a similarly high optimization benefit.

Table 3. Different Prediction Windows within Incremental Learning

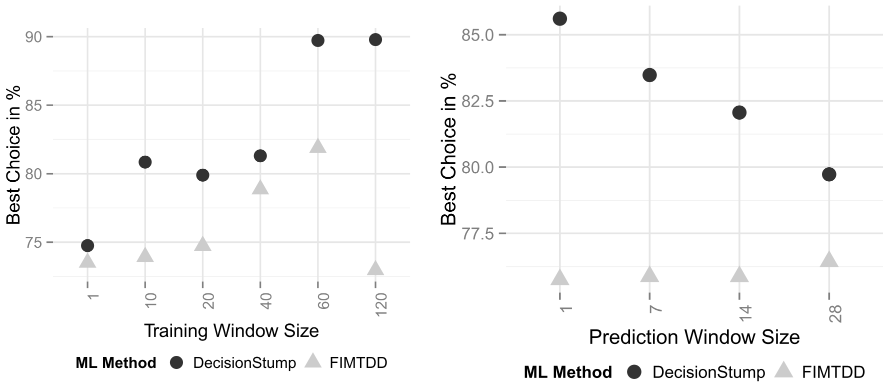
	DecisionStump				*FIMT-DD*			
Win. Size	**Achievement**		**Best Choice**		**Achievement**		**Best Choice**	
Prediction	*mean*	σ	*mean*	σ	*mean*	σ	*mean*	σ
1	97.10 %	6.10 %	82.26 %	22.89 %	97.04 %	3.89 %	73.35 %	21.47 %
28	96.34 %	2.23 %	72.77 %	22.94 %	97.02 %	1.60 %	72.78 %	13.59 %

Table 4. Sliding Window Scenario with Different Training Windows

	DecisionStump				*FIMT-DD*			
Win. Size	**Achievement**		**Best Choice**		**Achievement**		**Best Choice**	
Training	*mean*	σ	*mean*	σ	*mean*	σ	*mean*	σ
1	96.50 %	3.93 %	74.76 %	22.74 %	97.13 %	2.35 %	73.52 %	16.67 %
10	97.29 %	3.21 %	80.85 %	20.83 %	97.23 %	2.36 %	73.94 %	16.89 %
20	97.37 %	3.18 %	79.90 %	22.51 %	97.38 %	2.29 %	74.74 %	16.87 %
40	97.70 %	2.97 %	81.31 %	25.42 %	97.93 %	1.66 %	78.86 %	14.50 %
60	98.32 %	2.53 %	89.72 %	12.70 %	98.16 %	1.63 %	81.89 %	13.20 %
120	97.45 %	4.36 %	89.79 %	4.25 %	97.48 %	2.00 %	72.97 %	13.87 %

Table 5. Sliding Window Scenario with Different Prediction Windows

	DecisionStump				*FIMT-DD*			
Win. Size	**Achievement**		**Best Choice**		**Achievement**		**Best Choice**	
Prediction	*mean*	σ	*mean*	σ	*mean*	σ	*mean*	σ
1	97.86 %	4.81 %	85.61 %	17.27 %	97.48 %	3.55 %	75.76 %	21.43 %
7	97.42 %	3.52 %	83.48 %	17.33 %	97.54 %	1.95 %	75.87 %	16.27 %
14	97.31 %	2.91 %	82.06 %	18.18 %	97.56 %	1.53 %	75.87 %	12.92 %
28	97.16 %	2.22 %	79.73 %	19.52 %	97.63 %	1.16 %	76.44 %	10.71 %



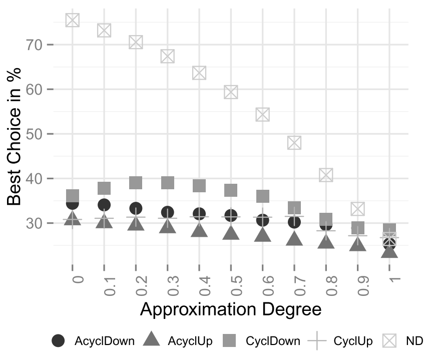
(a) Training Window Sizes (b) Prediction Window Sizes

Fig. 2. Best Choice Means of Different Window Sizes with Measurement Data

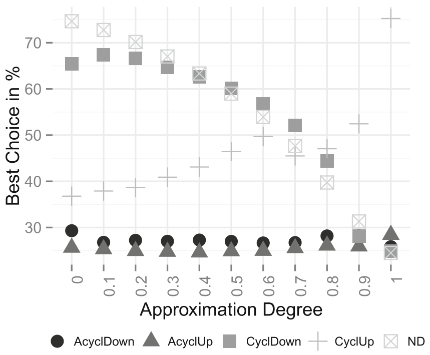
6.2 Simulated Scenarios

Based on the very good results of the measurement data and the results regarding the optimal window sizes for training and prediction, we now challenged both learning approaches with generated data. In contrast to the measured Web services with their quite distinctive characteristics, we generated the behavior of services with the simulation scenarios (profiles) introduced in Section 5.3. The services within a scenario have similar profiles but distinguish themselves in isolated focused aspects. Like the measurement data, the training on the generated datasets is also conducted on a sliding iteration basis. Additionally, we approximated each profile in 10 % steps until they were identical (disregarding some random noise) in order to address the last research sub-question. Presumably, the results of the machine learning approaches are supposed to get worse during the approximation. Still, the results of each approximation step reveal how good the learning approaches can cope with the challenge which the respective scenario focused on. The mean values in the illustrations were gained from the sliding window approach with a training window size of seven days and a prediction window of one day. This was the overall optimal combination for the measured real-world scenario. The generated input provided data for a period of six weeks in total. Since the simulated data followed a profile-guided generation, a longer period would not lead to different results in this case. Figure 3 depicts the best choice results for each machine learning approach. Figure 4 shows the correspondent overall achievement figures. Since the achievement is defined relatively between the best and worst service performances and since these performances are approximated step by step, there is not much difference between both figures with their different accuracy criteria "best choice" and "overall achievement", resp.; especially, when the approximation approaches 100 %. For the cyclic and acyclic profiles, the non-best services perform equally since there is no vertical shift. Hence, the overall achievement depends only on whether finding the best choice or not. Therefore, for these profiles, there is not much difference between the best choice and the overall achievement indicators.

Before we focus on the differences between both learning approaches, we compare the differences between the different scenarios. Both approaches cope well with the normal distribution scenario. This is the only scenario approximating a vertical shift (response time mean), and both methods and their approaches get worse when the response time means are approximated. All other scenarios approximate a horizontal shift. That means that their normal distribution component is and remains similar. They only distinguish themselves in their performance spikes (response time up for worse performance; response time down for improvements). In the acyclic spike scenarios, both approaches are not able to cope with these spikes. No matter whether the spikes go up or down, both approaches remain on a best choice rate of around 25 % which is not much better than random selection [3]. However, the DecisionStump approach achieves slightly better results. Comparing the remaining cyclic scenarios, the FIMT-DD can show its strengths (see CyclDown and CyclUp in Figure 3(b)). Compared to the classification-based approach illustrated in Figure 3(a), the FIMT-DD



(a) DecisionStump

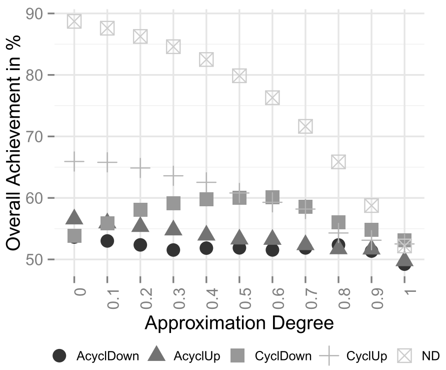


(b) FIMT-DD

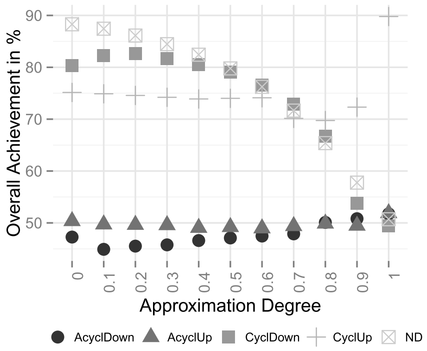
Fig. 3. Best Choice Mean using Sliding Windows within the Scenarios

achieves much higher best choice (and overall achievement) figures. The profiles of each service are taken into account, while this information is lost using the classification approach, which is illustrated in the charts in Figures 3 and 4. Our question, whether it makes any difference if the spikes go up (a service gets suddenly worse) or the spikes go down (a service gets suddenly better), could be answered. According to the results, illustrated in the figures, it does make a difference whether the spikes go up or down. FIMT-DD is in both cases significantly better. However, if a service gets suddenly worse among similar services (spikes down), it can be learned better than the other way around. It seems to be easier to learn an outstanding service whereas it seems to be more difficult to recognize a service getting worse within the optimization focus of similar well performing services. The presumed reason for that could be that spikes down (improvement of a single service) are changes within the optimization focus.

Having a closer look at the cyclic up illustration in Figure 3(b) and Figure 4(b), the indicator values get better with a higher approximation degree. This seems to be odd. One explanation could be that the regression-based approach focuses on the prediction of the performance behavior of each service as a pre-step for the actual best service determination, while the classification-based approach only focuses on the direct learning of the best fit service. However, the spikes up scenario simulates the opposite. Furthermore, considering the generation of the cyclic down and up scenario, their profiles are inverted on a higher level. The differences between the results in the figures also appear to be inverted. Still, the results for this scenario require further analysis, since a total approximation of this profile and its normal distribution part should develop similarly to the fully approximated normal distribution scenario.



(a) DecisionStump



(b) FIMT-DD

Fig. 4. Overall Achievement Mean using Sliding Windows within the Scenarios

6.3 Overall Results Conclusion

Both learning approaches achieved high overall optimization achievement results within the validation using real-world measurement data. Considering the best choice indicator, the classification-based approach could achieve higher values, however, with a higher standard deviation. Since real world-measured Web services have different profiles, which are easy to distinguish, challenging simulation scenarios showed the strengths and weaknesses of each approach. The regression-based approach for the prediction of the actual performance of each service candidate could prove its strengths in a cyclic behavior scenario. The consideration of a ranked determination of best services is especially beneficial since second-best service candidates can also still create an almost high optimization benefit. The normal distribution scenario showed that if the competitive service candidates mainly distinguish themselves vertically within the optimization focus, both machine learning approaches achieve equally good results.

6.4 Threats to Validity

The analysis is based on measured and generated data. Addressing an emerging future market, it is difficult to find freely consumable and functionally similar services. Therefore, the measurement data bases only on four Web services. The NFP behavior of Web services are diverse, therefore, the characteristics of other services may show different results. Nonetheless, with generated data of various scenarios as well as their approximation, we could analyze the strengths and weaknesses in detail. This would not be possible if the analysis is only based on measurement data.

7 Related Work

In [2], we introduced our service recommendation framework with an initial implementation using machine learning. We also determined appropriate machine learning frameworks which can be employed for service recommendation in general. First evaluations of the classification- and regression-based approaches were described in [3]. In these evaluations, we only had measurement input for 34 days. Furthermore, we did not conduct a sliding window approach. However, using different start, split, and end points can lead to different evaluation results. A sliding-conduction-based evaluation, which we used for the evaluation of the results of this paper, leads to more statistically profound results. In contrast to relative indicators (comparison to random selection), we used absolute indicators in this paper. Furthermore, in this paper, the generated simulation data focused specifically on general, presumable aspects of the performance behavior of services, which is beneficial for general evaluation of both approaches focusing on performance-behavior-based changes within a service market. Approaches using collaborative filtering (CF) for service recommendation also focus on the exploitation of shared knowledge about services in order to recommend services to similar consumers before the actual consumption on an automated basis (cf. [7–10]). Machine learning, in general, can also be used in CF. In contrast to the filtering of external decision results in CF, our approach determines the individual best-fit service based on previously measured performance data, individual preferences, and calculated utility values. With the call context and utility function approach, in our framework, new consumers can already benefit from existing knowledge. CF approaches also do not take into account that consumers can have different optimization goals or preferences and only some approaches [8,9] consider differences between consumers regarding their context. In [11], the authors tackle the lack of consideration of a consumer's preferences and interests; however, they do not take consumer context into account. The authors of [12,13] describe approaches to tackle the mentioned cold-start problem within CF. In [14], the authors used data mining methods for the service discovery. For the recommendation of services, trust and reputation are also important aspects. They can be understood in the security meaning, but also in a reliability context. For the latter, there are approaches which focus on a trust/reputation-based service recommendation [15–19]. Timelines within contexts are an important aspect. We leave this information to be handled by machine learning methods (apart from initial preparations). For detailed learning aspects on this context detail, we refer to the field of time series analysis.

8 Conclusion

With the sliding iterations of the input set, we conducted a statistically profound analysis of a classification- and regression-based machine learning approach for service recommendation. Both learning approaches achieved very high overall optimization achievement results within the validation using real-world measurement data. In contrast to the best choice indicator, for which both approaches also achieved high values, the overall achievement figure also takes into account the optimization achievement when recommending non-best-fit service candidates. The overall optimization achievement is more significant than the best choice indicator since it considers the actual utility/performance gain.

Considering the various training windows, the overall achievement figures remain similar when processing the measurement data. However, with an increasing training window size of up to 60 days, the best choice indicator improves for both approaches. For both approaches, the sliding window training strategy achieved better results than incremental learning. Although the classification approach achieved better best choice figures than the regression-based approach, both have similar overall achievements. So, the strength of the regression-based approach is the indirect ranking within the non-best services. Although the FIMT-DD regression method focuses on drift detection, a sliding training window achieved better results than the incremental learning approach.

In order to challenge both approaches and analyze their strengths and weaknesses in detail, we created profile-guided simulation data and re-conducted the analysis. Additionally, we approximated each generation scenario and re-run each analysis on each approximation step. For a normal distribution scenario, both approaches achieved similar good results. As expected, an increasing approximation degree resulted in less optimization achievement. In the acyclic case, both approaches did not achieve good results. However, the classification-based approach was slightly better. The regression-based approach achieved very good results within the cyclic profile scenario. Focusing on the best choice indicator, the approaches cope better with the scenarios with sudden performance improvements (spikes down) than with sudden performance decreases (spikes up). The unexpected improvements in the cyclic up scenario within the approximation steps require further analysis.

This work raised further future work questions: So far we have focused on a classification- and regression-based approach. Other approaches such as cyclic regression methods should be also evaluated. Also, other presumable behavior scenarios such as sudden death or continuously de-/increasing performance of services can be used for the analysis of the machine learning approaches. Furthermore, service recommendation also has an impact on the actual NFPs (e. g., NFP values of best-fit services might become worse due to over-consumption because of service recommendation), which also needs to be analyzed. In case of considerable negative effects, new strategies have to be found. Last but not least, strategies which aim at giving underdogs a chance to improve and quick starter strategies have to be found.

References

1. Andersson, J., Heberle, A., Kirchner, J., Löwe, W.: Service level achievements - distributed knowledge for optimal service selection. In: IEEE European Conference on Web Services (ECOWS) (2011)

2. Kirchner, J., Karg, P., Heberle, A., Löwe, W.: Appropriate machine learning methods for service recommendation based on measured consumer experiences within a service market. In: Int. Conf. on Advanced Service Computing (2015)
3. Kirchner, J., Heberle, A., Löwe, W.: Classification vs. Regression - Machine learning approaches for service recommendation based on measured consumer experiences. In: IEEE 11th World Congress on Services (SERVICES) (2015)
4. Ikonomovska, E., Gama, J., Džeroski, S.: Learning model trees from evolving data streams. Data Mining and Knowledge Discovery **23**(1) (2011)
5. Han, J., Kamber, M.: Data Mining: Concepts and Techniques, 2nd edn. Elsevier, Morgan Kaufmann (2006)
6. Kotsiantis, S.B.: Supervised machine learning: A review of classification techniques. Informatica (31) (2007)
7. Zheng, Z., Ma, H., Lyu, M., King, I.: QoS-aware Web service recommendation by collaborative filtering. IEEE Trans. on Services Computing **4**(2) (2011)
8. Tang, M., Jiang, Y., Liu, J., Liu, X.: Location-aware collaborative filtering for QoS-based service recommendation. In: IEEE Int. Conf. on Web Services (ICWS) (2012)
9. Kuang, L., Xia, Y., Mao, Y.: Personalized services recommendation based on context-aware QoS prediction. In: IEEE Int. Conf. Web Services (ICWS) (2012)
10. Yang, R., Chen, Q., Qi, L., Dou, W.: A QoS evaluation method for personalized service requests. In: Gong, Z., Luo, X., Chen, J., Lei, J., Wang, F.L. (eds.) WISM 2011, Part II. LNCS, vol. 6988, pp. 393–402. Springer, Heidelberg (2011)
11. Kang, G., Liu, J., Tang, M., Liu, X., Cao, B., Xu, Y.: AWSR: active Web service recommendation based on usage history. In: IEEE Int. Conf. on Web Services (ICWS) (2012)
12. Yu, Q.: Decision tree learning from incomplete QoS to bootstrap service recommendation. In: IEEE Int. Conf. on Web Services (ICWS) (2012)
13. Ahmed, T., Srivastava, A.: A data-centric and machine based approach towards fixing the cold start problem in web service recommendation. In: IEEE Students' Conference on Electrical, Electronics and Computer Science (SCEECS) (2014)
14. Nayak, R., Tong, C.: Applications of data mining in web services. In: Zhou, X., Su, S., Papazoglou, M.P., Orlowska, M.E., Jeffery, K. (eds.) WISE 2004. LNCS, vol. 3306, pp. 199–205. Springer, Heidelberg (2004)
15. Yao, J., Tan, W., Nepal, S., Chen, S., Zhang, J., De Roure, D., Goble, C.: Reputationnet: a reputation engine to enhance servicemap by recommending trusted services. In: IEEE Int. Conf. on Services Computing (SCC) (2012)
16. Zhang, J., Votava, P., Lee, T., Adhikarla, S., Kulkumjon, I., Schlau, M., Natesan, D., Nemani, R.: A technique of analyzing trust relationships to facilitate scientific service discovery and recommendation. In: IEEE International Conference on Services Computing (SCC) (2013)
17. Huang, K., Yao, J., Fan, Y., Tan, W., Nepal, S., Ni, Y., Chen, S.: Mirror, mirror, on the web, which is the most reputable service of them all? In: Basu, S., Pautasso, C., Zhang, L., Fu, X. (eds.) ICSOC 2013. LNCS, vol. 8274, pp. 343–357. Springer, Heidelberg (2013)
18. Li, L., Wang, Y., Lim, E.-P.: Trust-oriented composite service selection and discovery. In: Baresi, L., Chi, C.-H., Suzuki, J. (eds.) ICSOC-ServiceWave 2009. LNCS, vol. 5900, pp. 50–67. Springer, Heidelberg (2009)
19. He, Q., Yan, J., Jin, H., Yang, Y.: ServiceTrust: supporting reputation-oriented service selection. In: Baresi, L., Chi, C.-H., Suzuki, J. (eds.) ICSOC-ServiceWave 2009. LNCS, vol. 5900, pp. 269–284. Springer, Heidelberg (2009)