Optimized online learning for QoE prediction

Vlado Menkovski [a], Adetola Oredope [a], Antonio Liotta [a], Antonio Cuadra Sánchez [b]

[a] *Eindhoven University of Technology, P.O.Box 503, 5600MB Eindhoven, The Netherlands*
[b] *Telefonica I+D. 6 Emilio Vargas, 28043 Madrid, Spain*

Abstract

Quality of Experience (QoE) consists of a set of indicators that show the perceived satisfaction of using a multimedia (or other kind of) service by the end user. Being so the QoE presents a subjective metric and the only relevant mechanisms for measuring such indicators are subjective tests. Due to the fact that subjective tests are an expensive, impractical and in cases of live streaming a close to impossible exercise we set out on a twofold task to address this issue. First we set out to build prediction models using traditional Machine Learning (ML) techniques based on subjective test data. Second we explore an approach for reduction of the training dataset that will minimize the need for subjective data whilst keeping the prediction models as accurate as possible. For the first goal we used supervised learning based classification algorithms and we came up with high accuracy (over ninety percent) for the prediction models. To address the issue of high cost training data we developed a novel approach in reducing the training dataset while keeping a high accuracy of the classifiers. The reduction method provides a grading mechanism for unseen data. By having this mechanism in the online learning platform we can optimize the process of asking for user feedback by looking for the most significant cases, and therefore improving the gain on the trade-off between more feedback and more accuracy.

1 Introduction

With the advances in the telecommunication industry multimedia streaming services are increasingly more common. As with any service, the providers as well as the customers are interested in the level of quality of the service. Measuring the quality of multimedia services is not only important from the customer satisfaction point of view but also from the perspective of efficient management of the underlying network environment. In light of this, the importance of devising an effective measure of the quality of these services is of significant importance.

In this effort we are focusing on developing a mechanism for measuring the Quality of Experience (QoE) for multimedia services on mobile devices. The Quality of Experience consists of a set of indicators that show the perceived satisfaction of using the service by the end-user. These indicators include a vast variety of parameters from the multimedia encoding domain, transport as well as the type of terminal on which the media is presented and finally the type of content the user is watching. This QoE approach looks at the correlation of all these parameters to maximize the experience of the users while minimizing the resources of the provider.

QoE is a subjective measure because it depends on the perceived satisfaction of the viewer. Being so, the only relevant metric for comparison are subjective tests done with real users. But due to the fact that subjective tests are an expensive, impractical and in cases of live streaming a close to impossible exercise we set out on the task to build prediction models based on subjective test data for multimedia streaming services using Machine Learning (ML) techniques. Our goal is to be able to predict the QoE of the end users without having to do subjective test for each new multimedia contents and customer device.

In addition, to enable the adaptation of the prediction model to changes in user preferences we decided to use an online learning system. Realizing the cost of asking for user feedback we propose a method for guided reduction of the needed feedback data.

The prediction models that we have built using traditional ML techniques show accuracy of around ninety percent on a tenfold cross validation scheme. Further the Boundary Proximity method that we

propose shows that that with only a fraction (2%) of the data we can still maintain a reasonable prediction accuracy.

The rest of the paper is organized in the following manner. Section 2 discusses related work in QoE measuring. The next section discusses the Subjective test data and the prediction models trained on it. Section 4 discusses the Boundary Proximity method and finally Section 5 contains the conclusions of this work and proposed future work.

2 Related work

Most work around the area of QoE agrees with the definition that QoE is what the user perceives as quality while using a service [1]. Also it is generally accepted that QoE is a subjective measure and that a relevant comparison is with human subjective testing [2]. But it is also evident that subjective testing has a lot of drawbacks hence many efforts have been focused towards automating the QoE measurements most of which try to circumvent the subjective part with advanced objective techniques.

The ITU standardization process proposes five groups of objective models for measuring the QoE [1] of multimedia services. These groups range from Media-layer models, which look at the media content directly, then packet layer models that look at headers of packets, followed by parametric planning models that consider the network resource allocation, finally to bit-stream models and other hybrid models that combine different parts of the previous.

All of them address different aspects of the QoE. The Media-layer work is focused on the encoding and the effect of the compression on the fidelity of the multimedia content. It is shown that techniques as Mean Squared Error (MSE) and Peak Signal to Noise Ratio (PSNR) used here are lacking the understanding of the Human Visual System (HVS) due to which can deliver unsatisfactory results [3]. There are models that observe the content of the media and implement objective perceptual video quality measurement by modeling the HVS [3]. These models are computationally expensive because they need to execute in-depth analysis of the media content.

Looking only at transport or network conditions alone one overlooks the important parameters related to the content itself, the encoding process as well as the conditions in which the content is viewed. There are methods [4] that combine the Network Quality of Service (NQoS) and the Application Quality of Service (AQoS) towards achieving improved measurements in oppose to looking at parameters separately.

These efforts as well as all of the previous ones conclude that the single relevant reference metric to which the objective methods are measured, are the subjective tests. Looking at how the subjective tests are done in [5] and the obtained results we can observe the importance of taking into account external conditions like the type of the terminal and the content type to the perceived quality.

So to address the goal of looking at a correlation of all parameters and conditions and still provide an automated process we propose a system build by using ML techniques that predict the QoE. This approach is based on supervised learning on subjective test data, and some innovative design aimed towards minimizing the feedback needed for an online learning model.

3 QoE Prediction models

The prediction of QoE is based on classifiers trained on data from subjective tests that was done in [6]. The method used to design the subjective tests is known as Method of Limits [7]. It is used to detect the thresholds by changing a single stimulus in successive, discrete steps. A series terminates when the intensity of the stimulus becomes detectable. For the particular case we record the segment when the customer has decided that the multimedia quality is unacceptable. The purpose is to determine the user thresholds of acceptability for the particular QoS parameters taking into account the type of the content and terminal type. You can see an example of one test for the Mobile terminal on Table 1. The user was satisfied with the quality while the video bitrate was at or above 96Kbit/s. This example generates eight datapoints of which six are with a class label of Yes and two with No.

Segment	*Time (seconds)*	*Video bitrate (kbit/s)*	*Audio bitrate (kbit/s)*	*Frame rate*	*QoE*
1	1-20	384	12.2	25	Yes
2	21-40	303	12.2	25	Yes
3	41-60	243	12.2	20	Yes
4	61-80	194	12.2	15	Yes
5	81-100	128	12.2	12.5	Yes
6	101-120	96	12.2	10	Yes
7	121-140	64	12.2	6	No
8	141-160	32	12.2	6	No

Table 1. Example of a series of tests in the subjective study

The same tests are performed on different users showing them different video content as well as repeating the tests on three different terminals: mobile, laptop and pda. After compiling the results into three sets for each type of terminal we used the sets as training data for building prediction models. We used J48 an implementation of C4.5 [8] in the Weka platform [9] and SMO [10] an implementation of a Support Vector Machine also from Weka. The built models are shown in Figure 1 and Figure 2.

```
Video Framerate <= 6
|   Video Bitrate <= 32: no (106.0)
|   Video Bitrate > 32
|   |   Video SI <= 61
|   |   |   Video SI <= 50: yes (21.0)
|   |   |   Video SI > 50
|   |   |   |   Video SI <= 54: no (14.0/2.0)
|   |   |   |   Video SI > 54: yes (36.0/8.0)
|   |   Video SI > 61: no (30.0/2.0)
Video Framerate > 6: yes (647.0/40.0)
```

Figure 1a. C4.5 Model of the Mobile dataset

```
Video Framerate <= 12.5
|   Video Bitrate <= 32: no (103.0)
|   Video Bitrate > 32
|   |   Video SI <= 50: yes (21.0)
|   |   Video SI > 50
|   |   |   Video SI <= 67
|   |   |   |   Video TI <= 87: no (52.0/13.0)
|   |   |   |   Video TI > 87: yes (7.0)
|   |   |   Video SI > 67: no (24.0)
Video Framerate > 12.5
|   Video Framerate <= 20
|   |   Video SI <= 67: yes (172.0/15.0)
|   |   Video SI > 67
|   |   |   Video Bitrate <= 128: no (18.0/2.0)
|   |   |   Video Bitrate > 128: yes (21.0/7.0)
|   Video Framerate > 20: yes (436.0/1.0)
```

Figure 1b. C4.5 Model of the Laptop dataset

```
Video Framerate <= 10
|   Video Bitrate <= 32: no (105.0)
|   Video Bitrate > 32
|   |   Video SI <= 22: yes (32.0/9.0)
|   |   Video SI > 22
|   |   |   Video Framerate <= 6: no (96.0/9.0)
|   |   |   Video Framerate > 6
|   |   |   |   Video Bitrate <= 96: no (103.0/23.0)
|   |   |   |   Video Bitrate > 96
|   |   |   |   |   Video TI <= 93: yes (19.0/7.0)
|   |   |   |   |   Video TI > 93
|   |   |   |   |   |   Video TI <= 107: no (40.0/8.0)
|   |   |   |   |   |   Video TI > 107
|   |   |   |   |   |   |   Video TI <= 119: yes (9.0/2.0)
|   |   |   |   |   |   |   Video TI > 119: no (20.0/5.0)
Video Framerate > 10: yes (430.0/21.0)
```

Figure 1c. C4.5 Model of the PDA dataset

```
      1.4555 * (normalized) Video SI
+     1.0459 * (normalized) Video TI
+    -5.0892 * (normalized) Video Bitrate
+    -3.7632 * (normalized) Video Framerate
-     0.4582
```

Figure 2. SVM hyperplane for the Mobile dataset

```
      2.5405 * (normalized) Video SI
+     0.6061 * (normalized) Video TI
+    -4.2157 * (normalized) Video Bitrate
+    -4.3957 * (normalized) Video Framerate
-     0.7474
```

Figure 2b. SVM hyperplane of the Laptop dataset

```
      1.4229 * (normalized) Video SI
+    -0.4575 * (normalized) Video TI
+    -4.2913 * (normalized) Video Bitrate
+    -3.1618 * (normalized) Video Framerate
+     1.3385
```

Figure 2c. SVM hyperplane of the PDA dataset

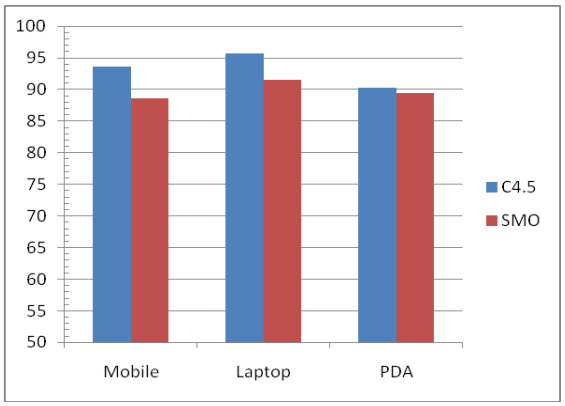

Figure 3. Comparison of accuracies for both models over the three datasets

Figure 3 shows the comparison the different accuracies for both models over the three datasets. Now that we have the ability to predict the QoE accuracy we can use these models in an online learning schema in order to create a flexible system that can adapt to the changes of user preferences, incorporation of new multimedia content and so on. A key component of the online learning loop is the user's feedback. From the previous discussion it is clear that subjective testing is costly and time consuming effort so now we are faced with the challenge to minimize the need of subjective feedback.

4 Minimizing user-generated feedback

In order to address the trade-off between the size of the dataset and classification accuracy we focused on uncovering a metric which will tell us the relevance of each datapoint. First we decided to set up a comparison method to measure the efficiency of any data reduction technique we develop further. For a comparison we look at the result of a random data reduction approach. In Figure 4 we can observe the dependency of the classifier's accuracy from the size of dataset. The random data reduction simulation was repeated one hundred times for each step in the data reduction and the values were averaged.

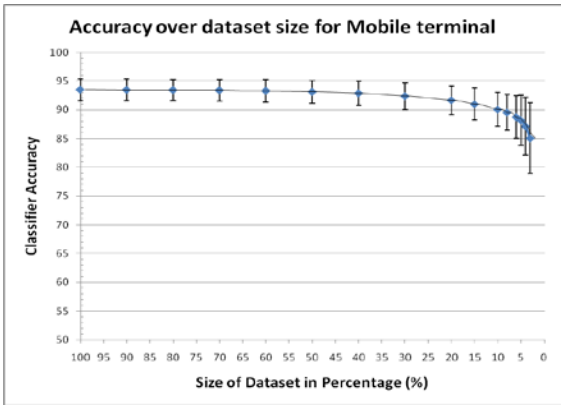

Figure 4a. Random Dataset reduction (C4.5)

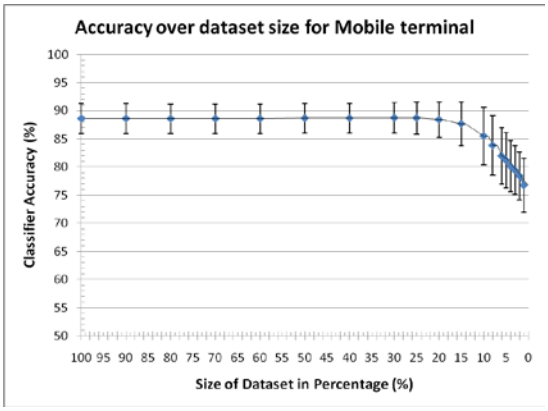

Figure 4b. Random Dataset reduction (SVM)

We can observe that the accuracy of the classifier falls to zero as it approaches the 0% mark for the dataset size. We also observe that the error bars (standard deviation) are increasing with the decrease of

accuracy, indicating that the given average accuracy is with lower confidence. In other words the random repeats produced classifiers with accuracy that varied further form the given average. These leads to the conclusion that there are some randomly selected subsets that deliver better results than others, but that it is also still possible to build a decent classifier that amount of data.
In light of this realization we want to find the pattern of data that trains the most accurate classifiers so we can only focus on this type of data and mange to decrease the size of the training set significantly while keeping the accuracy as high as possible.

4.1 Boundary Proximity Reduction Method

The intuition goes as follows. The data that is close to the decision boundary between one class value and another brings more precise information about the boundary itself than data further from it. So we need to focus on data that is in proximity to the boundary (Figure 5).

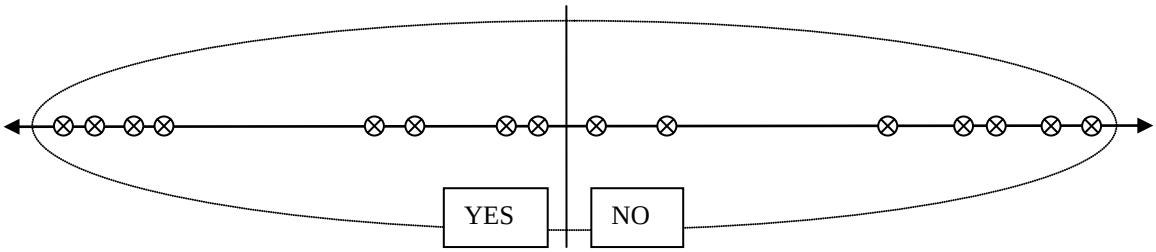

Figure 5a. Dataset projected on an axis representing the distance from the decision boundary

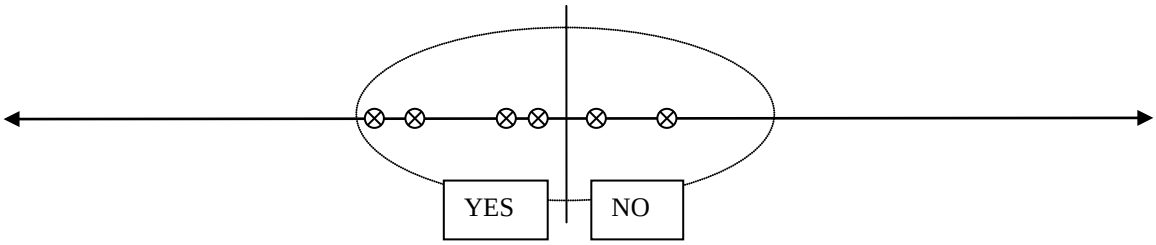

Figure 5b. Reduced dataset, preserving the datapoints close to the decision boundary

We expect that this approach will lead to a more stable dataset reduction and will deliver the means for a guided approach to data reduction. If the results show that a fraction of the data close to the hyperplane is enough to build an accurate classifier we can later use this approach to select the cases when we can ask for user feedback instead of asking randomly whilst getting much more useful data.

4.2 Experimental Setup and Results

To execute this experiment we built a Filter (Figure 6) for the Weka platform that reduces the data by a given percentage starting from the most distant datapoints. The filter works only in batch mode and cannot filter one instance at the time. This is due to the fact that it needs to build an SVM from the whole dataset initially. We extended the SMO algorithm so it will report the distance of each datapoint after the classifier has been built. Having this we now reduce the dataset starting from the datapoints with the largest distance from the boundary.

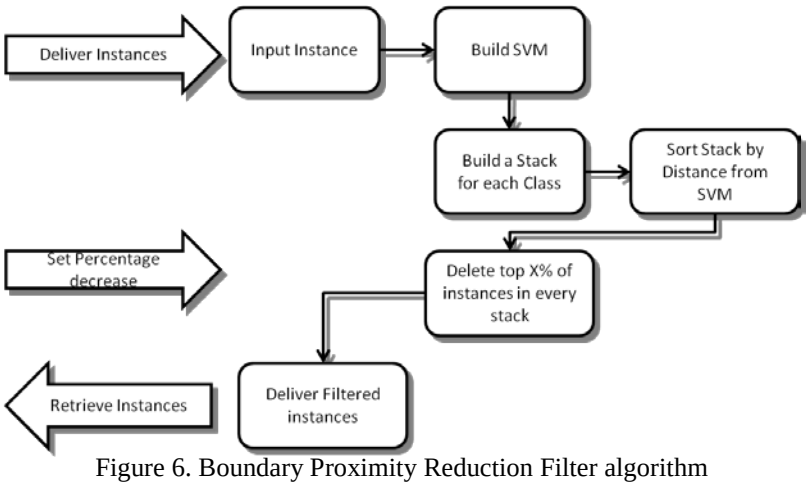

Figure 6. Boundary Proximity Reduction Filter algorithm

The results at this stage showed that the filter needs to be stratified in regards to label of the datapoints. This way the datapoints from each class or label will proportionally decrease in numbers. Otherwise we might disregard all the datapoints with a particular label while still having a lot of datapoints with the other labels. If that becomes the case the classifier's accuracy will be low since it will never see a case of that label during training time. After implementing a stratification of the filter the results that we got from experimental assessment over the mobile subjective dataset for the C4.5 decision tree are given in Figure 7.

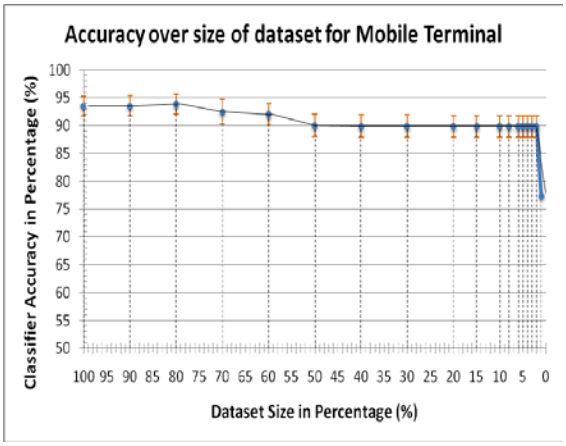

Figure 7. Accuracy of C4.5 on the reduced dataset

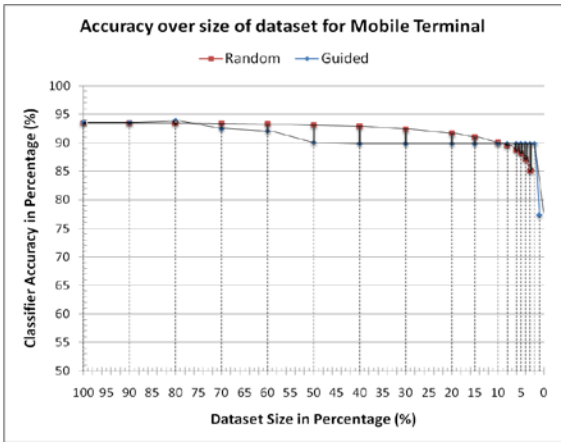

Figure 8. Overlay of both reduction methods

We can obeserve an overlay of both graphs on Figure 8. We see that the guided approach starts with the same accuracy as the random one and initially shows a slight improvement. Then the accuracy of the guided approach falls to 90% after the dataset has been reduced to 50% of original size, while the random avarage is slightly larger. This is due to the fact that SVM hyperplane that is used for the stratified boundary proximity reduciton is only 88% accurate as a classifier it self and does not capture all the exact information that is needed to build a more than 90% accurate treebased classifier. We will observe in the following discussion that the same is not true if we use a SVM as a classifier.

Where the approach shows its streanght is when we aggresivelly reduce the dataset to less than 10% of its initial size. We can observe that the random average falls bellow the guided approach. We can additionaly see that C4.5 performs with high accuracy (of arround 90%) right up until there is only 2% of the datset left.

If we observe the accuray when using an SVM classifier with the guided reduction we can see that the classifier accuracy is higly resistant to the redution process. Figure 9 shows that the classification

accuracy initially improves than stays fixed until the dataset size drops to 2%. We assume that the initial improvement is due to outlier removal from the datset, but this needs further investigatoin. On Figure 10 we can see a comparison between the guided and the random reduction. Here it is obvious that the guided reduction perfoms superiorly.

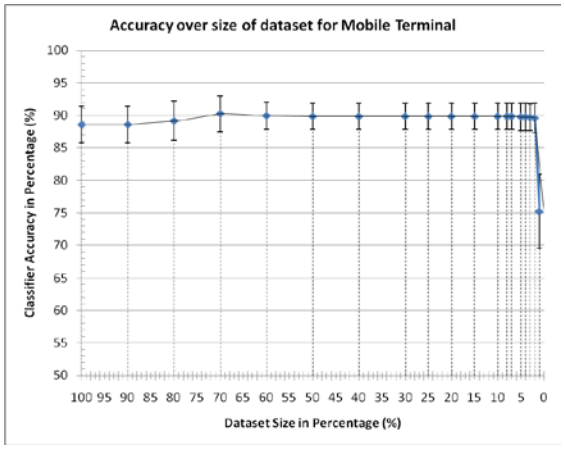

Figure 9. SVM classifier's accuracy on the boundary proximity reduction method

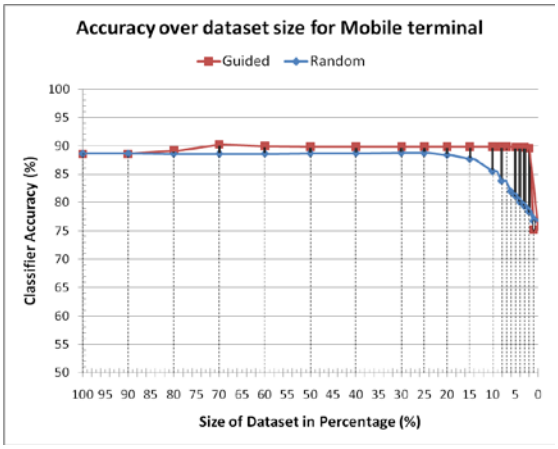

Figure 10. Overlay of SVM classifier's accuracy with random and guided reduction

Finally we compare the standard deviations in all four cases in Figure 11. We can observe that the deviation over the crossvalidation folds for SMO is intialy high, than after the reduction of outliers it falls and stabilizes bellow 2%. For C4.5 the standard deviation is generally stable. If we compare both of them to their random counterparts we observe that for random reduction of SMO the standard deviation from high goes to very high as the dataset is reduced. In addition for C4.5 the standard deviation from bellow 2% rapidlly increases to very high values as the datset is reduced.

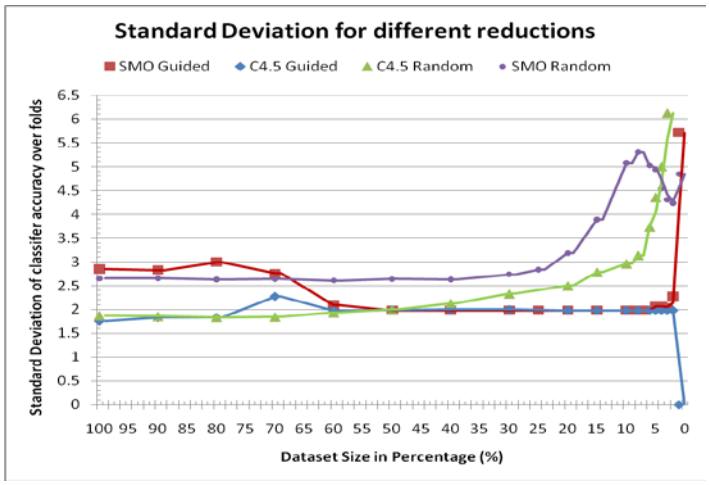

Figure 11. Overlay of the standard deviation of different reduction methods

For the dataset reduction analysis we have presented only tests over the mobile dataset for reasons of brevity. Results from the other datasets (laptop and PDA) are comparatively and qualitatively equivalent.

5 Conclusion and future work

In this work we have presented the importance of Subjective studies for measuring the QoE. Next we presented a means for the prediction of QoE by using models built with ML techniques, and finally we

have shown a method for reducing the dataset in order to optimize the tradeoff between the costs of acquiring data and maintaining an accurate online prediction model.

The results of applying traditional ML algorithms for building classifiers like C4.5 and SVM are excellent, yielding around ninety percent accuracy. Having those models we can confidently say that they are suitable for using in a QoE prediction platform with an online learning schema.

Next we set out to address the issue of reducing the need of feedback data for the online learning schema. By devising the Boundary Proximity Reduction method we showed an approach that offers a guided dataset reduction while keeping the accuracy of the classifier stable. It even shows that in our particular datasets it can perform outlier reduction and slightly improve the accuracy of the SVM. We also presented that the standard deviation for the classifiers during the reduction process stays stable and low (Figure 11) in oppose to the random dataset reduction.

These results are promising for building the online supervised learning QoE prediction platform. In order to generalize this approach, there is still need for further investigation of how the guided reduction method will perform on other datasets and other ML algorithms.

Acknowledgements

The work included in this article has been supported by Telefonica I+D (Spain). The authors thank all students and scientists who participated in these project activities. María del Mar Cutanda, head of division at Telefonica I+D and Florence Agboma for providing subjective QoE data from her PhD thesis work.

References

[1] A. Takahashi, D. Hands, and V. Barriac, "Standardization activities in the ITU for a QoE assessment of IPTV," *Communications Magazine, IEEE*, vol. 46, 2008, pp. 78-84.

[2] S. Winkler, *Video Quality and Beyond*, Symmetricom, 2007.

[3] S. Winkler, *Digital video quality : vision models and metrics*, Chichester West Sussex ;;Hoboken NJ: J. Wiley & Sons, 2005.

[4] M. Siller and J. Woods, "QoS arbitration for improving the QoE in multimedia transmission," *Visual Information Engineering, 2003. VIE 2003. International Conference on*, 2003, pp. 238-241.

[5] F. Agboma and A. Liotta, "Addressing user expectations in mobile content delivery," *Mobile Information Systems*, vol. 3, Jan. 2007, pp. 153-164.

[6] F. Agboma and A. Liotta, "QoE-aware QoS management," *Proceedings of the 6th International Conference on Advances in Mobile Computing and Multimedia*, Linz, Austria: ACM, 2008, pp. 111-116.

[7] G.T. Fechner, E.G. Boring, H.E. Adler, and D.H. Howes, *Elements of psychophysics / Translated by Helmut E. Adler ; Edited by David H. Howes [and] Edwin G. Boring ; with an introd. by Edwin G. Boring*, New York :: Holt, Rinehart and Winston, 1966.

[8] J.R. Quinlan, *C4.5*, Morgan Kaufmann, 2003.

[9] "Weka 3 - Data Mining with Open Source Machine Learning Software in Java."

[10] J.C. Platt, "Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines," 1998.