Ringtune: Learning-Based Context-Sensitive Implementation for Ringtone Volume Adjustment

Hsiang-Sheng Liang
(梁翔勝)
Department of Computer Science Information Engineering
National Taiwan University
r01922036@ntu.edu.tw

Jie-Wei Wu (吳潔薇)
Department of Electrical Engineering
National Taiwan University
r01921056@ntu.edu.tw

Yu-Cheng Huang (黃禹程)
Department of Electrical Engineering
National Taiwan University
b97901081@ntu.edu.tw

ABSTRACT

As technology advances, mobile phones become an essential part of our daily lives and makes instant communication much more convenient. However, there are occasions that we miss phone calls because of the vibration mode we forgot to turn off or we get embarrassed when the phone rings loudly during a class or in quiet places such as libraries. To cope with this problem, we implemented a learning-based context-aware application, Ringtune, that intelligently adjusts the volume level of mobile devices to the desirable level. Implementation guide lines and details on the algorithms used in Ringtune will be given.

1. INTRODUCTION

Smart applications such as Siri and navigation services provided by google emerge as mobile phones prevail in recent years, and all these can't be possible without the mature techniques in corresponding areas. Machine learning techniques have been widely studied in the past few years and several breakthroughs were made, intelligent learning models were proposed. With the observation described in the abstract, we propose Ringtune, a smart phone ring tuner that collects training data every time the user changes the volume, and fits perfectly with users in a customized way. The training data is used to train a classifier that predicts the desired ringer volume. When an incoming call arrives, Ringtune would turn off both the vibration and the ringer volume. After the features are extracted through the sensors and microphone, Ringtune sets the predicted ringer volume and the vibration is turned back on. State-of-the-art applications include RingDimmer [4], the best voted Android app for intelligent ring tunning. However, RingDimmer doesn't contain learning models, and it's possible that the rules built inside may go wrong and it'll never learn. Ringtune is a qualified ring tunner by the time this report is written, but our implementation still leaves room for the 'growth' of the classifier. Through incremental learning, Ringtune can hopefully match itself with the user's habits and customs.

This report is organized as follows. Section 2 contains related researches on volume changing of telephone and the related elements which might do the affect. The system designed with model and features extracted are listed and explained in Section 3. In Section 4, the implementation are described in detail. Achieved results are revealed in Section 5. Section 6 summarizes the report.

2. RELATED WORK

The idea that the mobile device should adapt to the surrounding environment originates even before current smart devices becomes popular. Siewiorek *et al.* [9] proposed a context-aware mobile phone. They attached a 3-axis accelerometer, light sensors and additional microphones to adjust the ringer volume and vibration of a feature phone via a predefined decision-making process.

Siewiorek's work is novel, but the empirical thresholds they used to classify their sensor data is a little primitive. Recent works leverage machine learning and better pre-processing for more meaningful features. Azizyan *et al.* [1] uses an 100-bin amplitude histogram to describe the ambience audio recording, in order to find out what kind of place the mobile phone currently situates. The histogram is compared to the existing entries in the database to filter out unlikely suggestions of location of the mobile phone. Welbourne *et al.* [10] classifies a mode of transit when the user moves by accelerometers. The component with 0.5 Hz 3 Hz is extracted and summed from the recorded accelerations by Goertzel algorithm. They chose the frequency band because previous work in biomechanics shows that the frequencies of interest for walking motions is below 10 Hz [11].

Finally, there are some existing mobile applications that changes the ringer volume adaptively [3, 4, 8]. How these apps are actually implemented is currently unknown. From the introduction they provided on Google Play, Let It Ring sets the ringer volume based on merely surrounding loudness [3]. RingDimmer records audio to listen to ambience noise and is also aware that the phone is in a pocket or not [4]. RingWise mutes the user's phone when an event in the calendar is currently on-going [8]. None of them claims that a machine-learning approach is taken, nor do these app gets smarter after the user uses it for a long time.

3. SYSTEM DESIGN

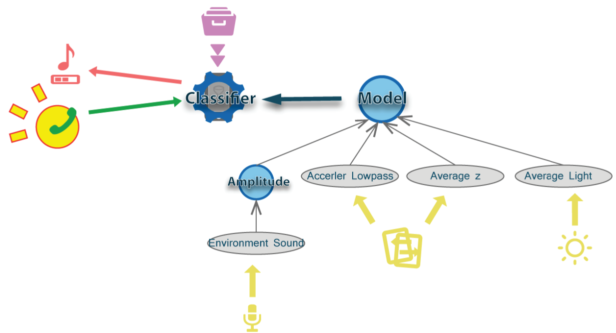

Figure 1: The structure of Ringtune ringer volume adjuster. An incoming call triggers the classifier to collect several features, and outputs the desired ringtone volume.

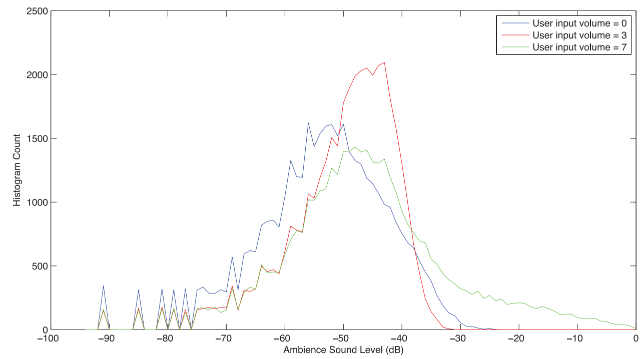

Figure 2: The amplitude histogram under different user input volume (label). Decibel (dB) unit is used to represent the ambience sound level.

Ringtune automatically adjusts the ringer volume based on the surrounding context it perceived and a classifier that contains two models. When an incoming call arrives, Ringtune mutes the phone temporarily, turns on its sensor for 1 second, processes the data and provides the data to the classifier. The whole process is depicted in Figure 1.

3.1 Sensor Features

During the 1-second sensory window, Ringtune would receive several values from one sensor, depending on the sampling rate of the sensor on the device. The several values from one sensor form a time sequence of data. The sequence of light sensor data, proximity sensor data and accelerometer data is used to produce the following features:

Average light
: The average of light intensity perceived (in luminance).

Average proximity
: The average of proximity boolean values. The sensor reports 1 in ‘near’ state and 0 in ‘far’ state. Thus the average proximity is a decimal between 0 and 1.

Average z-axis acceleration
: We observed that people often puts their phone on table with the screen faced up. In this situation the z-axis accelerometer would reports the acceleration for about 9.8 m/s^2, while it reports other values in other postures. This information would make a difference, hence this feature is provided to the classifier.

Low frequency acceleration
: We adopt how the acceleration values is processed in [10]. The FFT coefficients from 1 Hz to 3 Hz is simply summed up. The sum is used as a feature.

3.2 Ambience Sound as a Feature

Besides the sensor data, the ambience sound perceived by the microphone also plays an important role. However, the ambience sound contains too much information to be characterized by just a few features. Moreover, even if we managed to reduce the dimension of ambience sound to a small number of features, it does not make sense to mix up these ambience features with the sensor features.

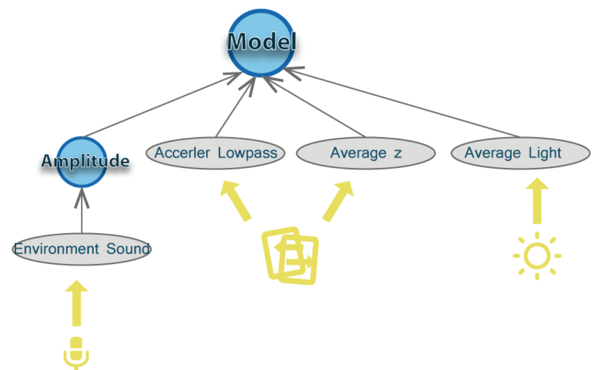

Figure 3: The hierarchical models of the Ringtune classifier. The ambience sound data is passed to the ambience model, which outputs an intermediate volume. The major model then predicts the ringer volume based on the intermediate volume and the sensor features.

The other model, the ‘ambience model’, is introduced to reduce the ambience features into one. The ambience model takes the amplitude histogram similar to the one in [1] and outputs an intermediate ringer volume as the label. The intermediate ringer volume is used as a feature for the major model.

The amplitude histogram used in Ringtune is slightly different from the one describe in [1]. After inspecting the collected ambience sound data, we found that when a logarithmic unit such as Decibel is used in representing the sound level (Figure 2), the amplitude histogram between different labels differs more. This makes it easier for the ambience model to classify the histogram into different labels. 95 sound level bins are used in the histogram, ranging from 0 dB to -95 dB.

3.3 The Two Models

As mentioned above, the Ringtune classifier (Figure 3) has two models involved in determining the final ringer volume.

The ambience model takes the amplitude histogram as input and outputs the intermediate ringer volume. The major model takes in the 4 sensor features and the intermediate ringer volume as input, and outputs the final ringer volume.

The implementation of the two models is explained in detail in the next section.

3.4 Training Data Collection

We collect the features and the ringer volume settings as training data in the daily use of the mobile phone. If the user 1) adjusted the volume manually, or 2) received an incoming call, Ringtune automatically collects the feature and records the ambience audio 5 seconds after the user turns off the screen of the phone. We assume that under these circumstances, the user should be satisfied with the current ringtone volume settings. The 5 second delay after switching off the screen is to make sure the user have put their phone down when Ringtune starts collecting data. Reason is that we intend to collect data of the practical situation the incoming call really encounter.

4. IMPLEMENTATION

Core objects and environment/version information in Ringtune are explained here.

`MachineLearningController` is the core object in Ringtune, for it controls the volume label when a phone call occurs, and records all the adjustments made by the user. Incremental learning is left for future work. The current version takes a trained model from either Weka GUI or any other ways that produces a Weka model file. It's worth mentioning that Ringtune adopts a two-layered modeling for our classification scheme, which consists of an ambience model and a major model. (Please refer to section 3.3) The ambience model converts a recorded sound file (e.g. mp4) to a fixed-sized feature vector, and the major model takes the converted feature vector along with other attributes computed from sensor data (e.g. average light intensity) as its input. The major model therefore outputs the goal value, i.e. the suggested phone volume for `VolumeManager` to set the system volume.

`RingtuneLogger` is the interface to SQLite database and the server in `parse.com`. It archives all data to the local database and uploads them all to parse.com whenever there is available Internet/Wifi around.

The Weka library is imported into Ringtune, and it stores all archived changes (training data) to the built-in SQLite database. Ringtune runs on Android API level 15 or higher, and Weka-3-6-8 is used for building models and classification. The models included in Ringtune are SMO (Sequential Minimal Optimization) [2, 5], KNN (K-Nearest Neighbour), ANN (Artificial Neural Network) [7], and J48 (Decision Tree) [6], and most of them are from Weka-3-6-8 library, except the KNN algorithm, which is implemented by ourselves for special purposes on dealing with environment noise/sound levels.

5. ACHIEVED RESULT

At the end of this semester, Ringtune has achieved what it takes to be a smart ring tuner in a sense of functionality. Ringtune is now capable of archiving training data by sampling user behavior, that is, any volume adjustments along with the environment data will be stored into database and uploaded to a centralized server site parse.com for research purposes, and the data are ready for incremental learning, which, however, is yet to be implemented in the future. At the moment this report is written, we used a model trained out of the centralized data for picking a proper learning algorithm.

With J48 being the major model and SMO being the ambience model, Ringtune can reach an accuracy of 72.10%. In this model combination, we found the decision tree this combination yields makes some sense. As Figure 4 suggests, the volume output tends to be 0 or 1, which represents a vibration mode or the smallest phone volume, when the sound level (environment noise) is low, and the light is dimmed. (The left-most branch on Figure 4) It's easy to come up with an idea that the user is probably sleeping, which makes perfect sense. Take the right-most extreme case for example, when the sound level is hight, and the light is dimmed, it outputs the highest volume possible for an android phone, 7 to cope with the extremely noisy environment, for that dimming light suggests that the phone might be in a pocket or a backpack, and under such circumstances a high sound level would definitely be a clear sign to noisy environment.

With the above observations in mind, it is natural that, when equipped with more training data, Ringtune is going to reach a much higher accuracy and makes a perfectly smart ring tuner that deals with customized habits.

6. CONCLUSION

In this project we implement an app that uses the sensors to collect contextual data of an incoming phone call. It then decides the best ringer volume with a classifier based on the training data that encodes the user's past behavior.

Light intensity, proximity, magnitude of acceleration and ambience sound fingerprint are taken into consideration by the classifier. The classifier is composed of two models, the major and the ambience model, which are illustrated in Figure 3. Combinations of different models have been thoroughly investigated, including decision tree, Support Vector Machine, K-Nearest Neighbour and artificial neuro-network. The co-operation of decision tree and the support vector machine gives the best performance.

However, our experiments obviously suffers from the skewed data. This app is designed to *grow* with the user. That is, the more data from the user, the smarter the app is. Currently, the scarcity of data provides little support to infer any useful knowledge. We believe that as the amount of data increases, there is still a great growth potential in our incremental learning.

Moreover, more user information can be taken into consideration. For example, if the app can read the user's Google calendar, some inferences can be made directly from the schedule and thus increases the learning capability of the app. Chances are that the app would perfectly fit the habits and customs of the user. By then the app would be much more practical and convenient for the user herself/himself. Hopefully in the future we no longer suffer from missing calls and embarrasing loud ringtone volume.

7. REFERENCES

[1] M. Azizyan, I. Constandache, and R. Roy Choudhury. Surroundsense: mobile phone localization via ambience fingerprinting. In *Proceedings of the 15th*

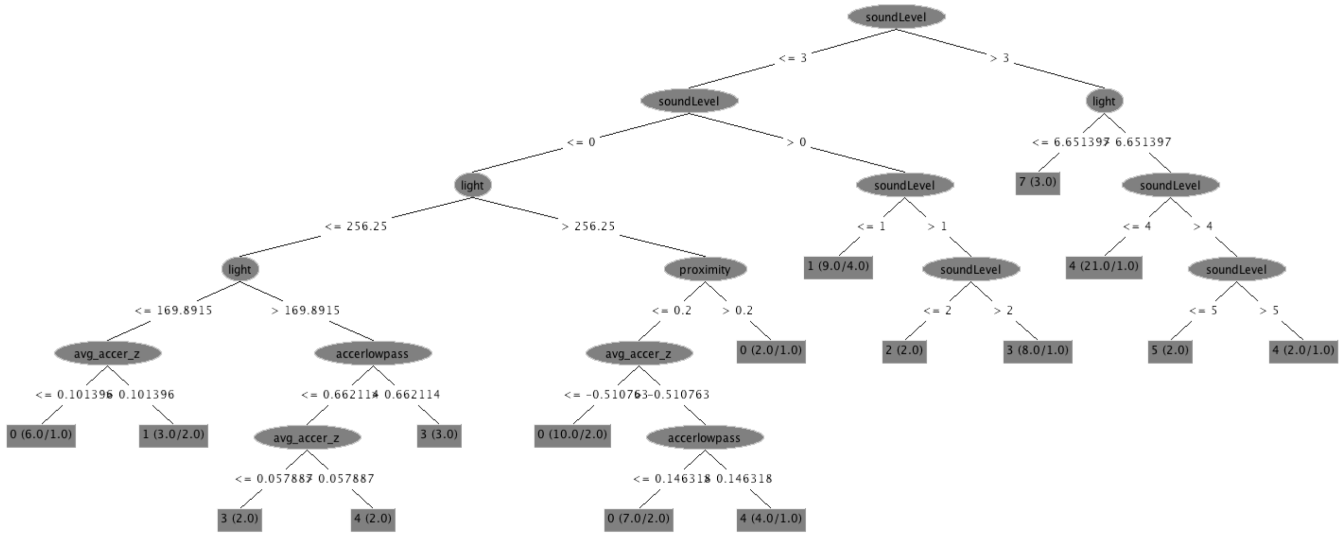

Figure 4: The above picture shows the decision tree built by the model combination consisting of J48 (major model) and SMO. (ambience model)

annual international conference on Mobile computing and networking, MobiCom '09, pages 261–272, New York, NY, USA, 2009. ACM.

[2] S. S. Keerthi, S. K. Shevade, C. Bhattacharyya, and K. R. K. Murthy. Improvements to platt's smo algorithm for svm classifier design. *Neural Comput.*, 13(3):637–649, Mar. 2001.

[3] T. Mutators. Let it ring. `https://play.google.com/store/apps/details?id=com.techmutators.letitring`. Accessed Jan 22, 2013.

[4] M. Pardo. Ringdimmer. `https://play.google.com/store/apps/details?id=com.michaelpardo.ringdimmer.free`. Accessed Jan 22, 2013.

[5] J. C. Platt. Advances in kernel methods. chapter Fast training of support vector machines using sequential minimal optimization, pages 185–208. MIT Press, Cambridge, MA, USA, 1999.

[6] J. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Series in Machine Learning. Morgan Kaufmann Publishers, 1993.

[7] R. Reed and R. Marks. *Neural Smithing: Supervised Learning in Feedforward Artificial Neural Networks*. A Bradford book. Mit Press, 1999.

[8] B. Richey. Ringwise. `https://play.google.com/store/apps/details?id=com.behzad.ringwise`. Accessed Jan 22, 2013.

[9] D. Siewiorek, A. Smailagic, J. Furukawa, A. Krause, N. Moraveji, K. Reiger, J. Shaffer, and F. L. Wong. Sensay: A context-aware mobile phone. In *Proceedings of the 7th IEEE International Symposium on Wearable Computers*, ISWC '03, pages 248–, Washington, DC, USA, 2003. IEEE Computer Society.

[10] E. Welbourne, J. Lester, A. LaMarca, and G. Borriello. Mobile context inference using low-cost sensors. In *Proceedings of the First international conference on Location- and Context-Awareness*, LoCA'05, pages 254–263, Berlin, Heidelberg, 2005. Springer-Verlag.

[11] D. Winter. *Biomechanics and motor control of human movement*. A Wiley-Interscience publication. Wiley, 1990.