AudioSense: Enabling Real-time Evaluation of Hearing Aid Technology In-Situ

Syed Shabih Hasan†, Farley Lai†, Octav Chipara†, Yu-Hsiang Wu‡
† Department of Computer Science/Aging Mind and Brain Initiative
‡ Department of Communication Sciences and Disorders
University of Iowa
{*syedshabih-hasan, poyuan-lai, octav-chipara, yuhsiang-wu*}@uiowa.edu

Abstract

AudioSense integrates mobile phones and web technology to measure hearing aid performance in real-time and in-situ. Measuring the performance of hearing aids in the real world poses significant challenges as it depends on the patient's listening context. AudioSense uses Ecological Momentary Assessment methods to evaluate both the perceived hearing aid performance as well as to characterize the listening environment using electronic surveys. AudioSense further characterizes a patient's listening context by recording their GPS location and sound samples. By creating a time-synchronized record of listening performance and listening contexts, AudioSense will allow researchers to understand the relationship between listening context and hearing aid performance. Performance evaluation shows that AudioSense is reliable, energy-efficient, and can estimate Signal-to-Noise Ratio (SNR) levels from captured audio samples.

1 Introduction

A 2008 MarkeTrak survey estimates that 11.3% of Americans (approximately 34.25 million) suffer from hearing loss [11]. Hearing loss often leads to social isolation that has significant deleterious effects on one's health. For example, hearing loss in older adults has been associated not only with communication difficulties, but also with decreased health and reduced engagement in physical activities [2]. The primary intervention for sensorineural hearing loss and related psychosocial consequences is hearing aid amplification. However, in spite of significant advancements in hearing aid technology during the past decade, hearing aids use is not prevalent among people with hearing loss [9, 11] and only half of those using hearing aids are satisfied with their performance in noise [7]. Moreover, several recent clinical studies indicate that laboratory assessments of hearing aid performance are not predictive of their real world performance [1, 10, 13, 14]. Therefore, in order to improve hearing aids, there is a critical need to develop assessment techniques that allow engineers and clinicians to understand the factors that affect hearing aid performance in the real world.

Measuring the performance of hearing aids in the real world poses significant challenges as it depends on the patient's *listening context* which includes characteristics of listening partners, listening activities, location of conversation partners, and environment. Audiologists currently measure hearing aid performance either through self-reporting methods or speech-in-noise tests. Self-reports are commonly used to assess the auditory handicap and patient satisfaction with hearing aid performance. Unfortunately, self-reports are plagued by memory biases, as patients are required to remember the circumstances in which hearing aids performed poorly long after they occurred. Speech-in-noise laboratory tests are used to assess the benefits of hearing aids, configure parameters of amplification algorithms, and compare different hearing aid technologies. During a test, a patient placed in a sound booth is presented segments of speech under different noise conditions. As these tests are usually focusing on showcasing the various aspects of hearing aid technology (e.g., use of omnidirectional vs. directional microphones) they fail to be representative of the listening contexts that patients encounter during their daily life. Accordingly, neither self-reporting nor speech-in-noise tests are effective in describing the listening contexts observed by patients in the real world.

In this paper, we present AudioSense, a novel system for evaluating hearing aid performance in the real world that integrates mobile phones and web technology. The novelty of AudioSense is that it *combines subjective and objective measures of hearing aid performance and listening contexts*. AudioSense uses Ecological Momentary Assessment (EMA) methods. EMA involves the repeated sampling of the subject's current state and experiences in real-time [12]. We use EMA to evaluate both the perceived hearing aid performance as well as to characterize the listening environment (e.g., listening activity, room size, and location of speakers). This is accomplished by delivering electronic surveys either at randomized intervals or when triggered by patients. Compared to other self-reporting methods, EMA has the advantage of reducing memory bias since patients report on their recent experiences (in the previous 5 - 10 minutes). Concurrently with the delivery of surveys, AudioSense further characterizes a patient's listening context by recording their GPS location and sound samples. Standard sound analysis techniques (e.g., computing SNRs) are used to analyze the sound samples after upload to a web server. GPS locations could be used to determine whether the subject is indoors or outdoors. By creating a time-synchronized record of listening performance and listening contexts, AudioSense opens significant opportunities to understand the relationship between listening contexts and hearing aid performance.

The implementation of AudioSense has been evaluated across three dimensions: reliability, energy consumption, and errors in SNR estimation. Experimental results indicate that 100% of the surveys were successfully collected in spite of intermittent network connectivity. Moreover, AudioSense can deliver surveys at 1.5 hours intervals for two days without requiring the mobile phone to be recharged. Finally, we have evaluated the ability of estimating SNR from sound files when various levels of Gaussian noise were added. Preliminary results indicate that the average SNR estimation error was 0.62 dB.

2 Related Work

EMA has been proposed as an alternative to retrospective self-reporting methods that suffer from memory bias. A PubMed literature search indicates that only two audiology studies have used computer-based EMA to date. Henry et al. [5] used EMA to evaluate the impact of chronic tinnitus[1] on the day-to-day activities of patients. Galvez [4] used EMA to evaluate patient satisfaction with hearing aid performance. In contrast to the tools used in these studies, AudioSense can track of patient compliance in real-time using a web portal. Galvez reports a compliance rate of 77% in his study. We expect that by tracking patient compliance in real-time, AudioSense may achieve higher compliance rates. More importantly, neither study collects any sensor data to characterize the patient's context.

While audiologists continue to use relatively simple versions of EMA, computer scientists have proposed to combine experience sampling and collection of sensor data to capture contextual information [3, 6]. However, clinicians have not adopted these techniques since they do not include domain-specific measures of contextual information that are necessary to assess their medical relevance. AudioSense addresses this limitation by providing an extensible environment for using algorithms for characterizing the listening context.

[1]Tinnitus is the perception of sound in the ear and may interfere with hearing.

Speech-in-noise tests are widely used to assess the benefits of hearing aid noise reduction technologies. Such tests including QuickSIN and Hearing in Noise Test (HINT) present speech and noise at different SNRs. Among the contextual factors that would affect hearing aid users' speech understanding, SNR is probably the most important one. AudioSense already includes algorithms to characterize the SNR of collected speech. In the future, we plan to integrate AudioSense with other algorithms to further classify and characterize listening contexts. We will leverage on the significant body of work on sound classification (e.g., [8]); many of such algorithms are already implemented in MATLAB allowing for a simple integration with AudioSense.

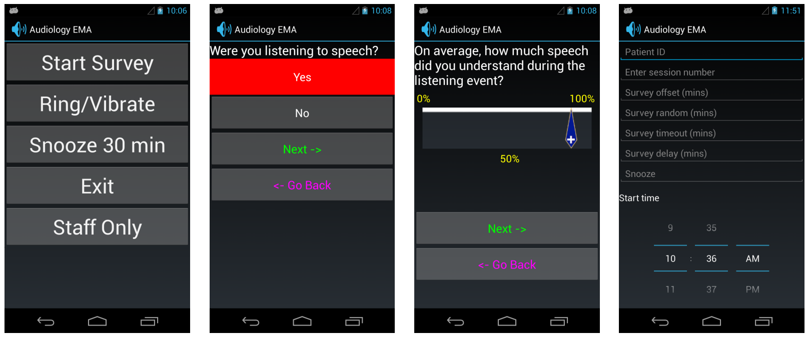
3 AudioSense System

AudioSense is designed to collect objective measures of hearing aid performance and listening contexts in the real world. The design of AudioSense must address four key requirements: (1) must facilitate compliance with data collection protocols over multi-week deployments, (2) must ensure the reliability of data collection, (3) must provide an extensible software architecture to enable signal processing and audio analysis on collected sensor measurements, and (4) support concurrent data collection from multiple users. In the following, we present the system architecture and software components of AudioSense, focusing on how the system addresses these requirements.

3.1 System Architecture

AudioSense is a two-tier system that is composed of mobile phones and a backend server. The mobile phones are carried by patients and are used to deliver surveys and collect sensor measurements. The server backend includes three components: a web server, a database, and a speech analysis component. The web server stores the data uploaded by clients in a database. The web server provides a standard web portal interface to visualize the collected data and monitor patient compliance with data collection regiment. The speech analysis component allows the uploaded data to be automatically processed in the MATLAB environment. We opted to integrate with MATLAB to provide a flexible and extensible environment for signal processing and speech analysis. This choice is motivated by the availability of several speech analysis algorithms as open-source components implemented in MATLAB (e.g., VoiceBox).

The communication between mobile phones and the web server is accomplished using HTTP over Wi-Fi or a cellular network. As patients in our studies are mobile and may live in rural parts of Iowa, wireless connectivity may be intermittent. AudioSense is designed to tolerate intermittent network connectivity by having each mobile phone cache the collected data aggressively. Periodically, the mobile phone attempts to establish connections to the web server and, when successful, it uploads the collected data. Note that the storage space available on modern mobile phones



(a) Home screen (b) Multiple choice (c) Scale rating (d) Settings screen

Figure 1. EMA component

is sufficient to store all the data that we collect even in a multi-week deployment.

3.2 Software Components

The client-side of AudioSense runs on mobile phones and is implemented on top of Android OS. Android OS is available on numerous mobile phones and tablet computers. AudioSense can be deployed on any Android device. The backend server is portable and can be deployed on Mac OS, Linux, and Windows. The web portal is implemented using the Django web framework. SQLite is used to store data and manage metadata associated with the collected sensor readings and surveys. MATLAB is used as a computing environment for analyzing collected sensors measurements. Next, we describe each software component.

3.2.1 EMA Component

The EMA component runs on mobile phones and is responsible for managing activities associated with the delivery of electronic surveys. The EMA component addresses the needs of both software developers and patients.

A software developer can create new surveys using a simple API. A survey is modeled as a set of questions. To keep track of the patients' choices at run-time, we associated with each question a variable to which we assign a value based on the response of the patient to each question. A patient may navigate through the survey both forwards and backwards. They may revise their answers as necessary. The next question presented to the patient depends on his previous answers, thus allowing for adaptive surveys.

While the EMA component has an extensible architecture, we currently support two types of questions: multiple-choice questions and scale rating. Multiple-choice questions are rendered as a sequence of buttons whose text can be specified by the programmer (see Figure 1(b)). The patient is allowed to select a single option out of those presented. Scale rating questions are rendered using seekbars and the programmer can provide labels to be rendered for the middle and ends of the bar (see Figure 1(c)).

The delivery of electronic surveys may be alarm triggered or patient-initiated. The EMA component supports the delivery of surveys using either fixed or a randomized schedules. If a survey was just delivered, the time offset until the next survey will be delivered is computed by adding a constant time offset T_{offset} and a random number picked uniformly from the time interval $[0, T_{rand}]$. This method allows for the generation of both fixed schedules (i.e., by setting $T_{rand} = 0$) as well as randomized schedules. Typically, our audiology surveys are delivered on average every 1.5 hours and consecutive surveys are separated by at least 1 hour (i.e., T_{offset} = 1 hr and $T_{rand} = 1$ hr). Moreover, in order to minimize the interruption burden to patients, clinicians can select the time interval during a day when surveys can be delivered. An alarm outside the delivery interval will be postponed until the next day.

Appropriate user interface design can have a significant impact on the compliance of patients with the data collection protocols. This is particularly problematic given that patients with hearing loss also tend to be older. Accordingly, they do not only suffer from hearing loss but also may have impaired vision and potential loss of fine motor control. These considerations influenced our user interface designing choices. We refined our initial user design based on patient feedback. Accordingly, we opted for large font sizes and a color scheme whose colors are easy to distinguish. Similarly, we opted for a large buttons and overrode the default seekbar provided by Android OS with one that provides a larger area that is sensitive to touch. The most consequential decisions in the user interface are related to the delivery of alarms – notifications that the user should complete a survey. After several iterations and feedback from patients, we decided to deliver survey alarms by vibrating the phone, playing loud ringtones, and turn on/off the flash of the camera. An alarm sounds for 30 seconds. Our choice for an alarm that can be quite intrusive and irritating is balanced by the ability to easily dismiss it: the patient may press the power button to stop the alarm. Moreover, we have added a *Snooze* option that allows the patient to postpone completing the survey by 30 minutes.

3.2.2 Sensor Data Collection

While surveys are administered, AudioSense records audio at 16 KHz and GPS locations at 0.1 Hz. The data collection is triggered either by an alarm or when the user opens the application. The data collection is stopped after a timeout configured by the developer.

Unlike the EMA component that utilizes only a fraction of the phone's resources, the design of sensor data collection must minimize resource utilization. To this end, AudioSense implements a simple but effective pipeline abstraction: in a pipeline, the data flows from the source to the sink and is transformed by the intermediary components. Each pipeline is executed in a different thread in order to isolate the data collection from different sources. An additional concern is the need to minimize the number of times the garbage collector is invoked on Dalvik Virtual Machine (DVM). Each time when the garbage collector identifies objects that are no longer used by an application it reclaims the allocated memory. The garbage data collection operation interrupts the execution of the application between 10 – 100 ms depending on the number of objects freed. While most applications would not be affected by this delay, when high rate audio is recorded, such delay may lead them to drop audio frames. We ensure that the objects used in data collection never need to be garbage collected using the following approach. Each pipeline sources manages a shared buffer pool that contains a number of frames preallocated when the application starts. When a source has data to write, it retrieves a frame from the buffer pool and writes the data into the frame. Frames are pushed down the pipeline through each intermediary component, which receives a reference to that frame. Upon reaching a sink, the frame is put back into the buffer pool of its source. This mechanism of cycling the frames between sources and sinks prevents the frames from being garbage collected since they are always in use.

AudioSense includes three pipelines: audio processing, GPS processing, and file upload. The audio and GPS pipelines have a similar behavior: they collect data from their respective sensor source and save it to a file. Upon the completion of the data collection, the names of files containing the sensor data are passed to the file upload pipeline. The file upload pipeline maintains a queue of the files that are to be uploaded. The content of the queue is saved to disk in order recover from application crashes without losing information according to the following policy. When a new file is added to the queue, the content of the queue is saved immediately to disk to avoid data loss in case of an application crash. In contrast, when a file is removed from the queue, this operation does not result in an immediate write to disk as in the worst case this would lead to a file being uploaded twice. The file upload pipeline dequeues the names of the files and creates HTTP POST request to be sent to the server that includes the file and additional metadata. The metadata includes a patient identifier, a phone identifier, a session identifier, and the time when the data was collected. Upon a successful upload, the uploaded file is removed from the queue.

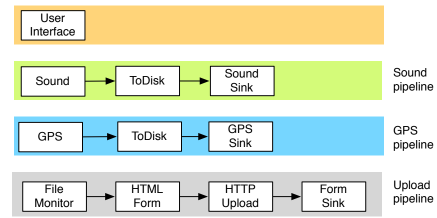


Figure 2. AudioSense: Client implementation

AudioSense uses the power-lock interface provided by Android OS to manage its power usage. The EMA component acquires a lock that maintains an active screen at the start of a survey. If the EMA component does not receive any user input for one minute, the survey component is stopped and the screen power lock released. This indicates to the OS that it may turn off the screen if no other application has acquired a power lock on the screen. AudioSense maintains a CPU lock during the collection of sensor data. During the delivery of alarms AudioSense also turns on the camera to access the flash, but it turns it off after the 30 seconds alarm is delivered. In a typical deployment, AudioSense is, on average, active for 10 minutes every 1.5 hours resulting in an 11.11% duty cycle.

3.2.3 Web Server Backend

The AudioSense web application is implemented using the Django web framework. Django provides basic facilities for secure website login and user management. The AudioSense web application takes advantage of these capabilities to provide a simple user portal. The primary goal of the user portal is to provide clinicians access to real-time data for monitoring patient compliance.

The web application is responsible for handling the HTTP POST requests from clients. Each HTTP request includes the actual data along with identifiers for the patient and phone where the data was recorded. The metadata is stored in a database for easy querying while the files are stored on the local hard drive. For security purposes, the local hard drive is encrypted. A request also results in a new processing job being added to speech analysis component. The web server may serve multiple clients concurrently.

The speech analysis component integrates with MATLAB environment on the server. This allows AudioSense to be an extensible environment in which many backend algorithms can be implemented. Currently, we have implemented a number of algorithms for estimating the SNR from collected speech segments. Our focus on SNR is justified by the fact that it is a good indicator listening context. SNR computations may be triggered when an upload is completed.

4 Performance Evaluation

The key to successfully deploying AudioSense is to ensure reliable and energy efficient data collection. Accord-

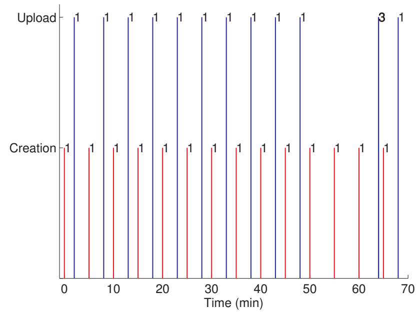


Figure 3. Reliability

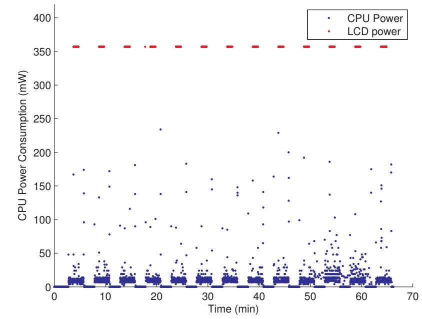


Figure 4. Power Consumption

ingly, this section measures the reliability and power consumption of AudioSense under a realistic deployment scenario. These results are complemented by preliminary results from actual field deployments. Additionally, we also evaluated AudioSense's capability of estimating SNR using the MATLAB backend.

We configured AudioSense to deliver surveys every five minutes. AudioSense operated as follows: during the first three minutes of each data collection round, AudioSense recorded sound samples and GPS locations. One minute within each data collection round, AudioSense triggered an alarm for the user to complete the surveys. During the experiments, AudioSense recorded sound and GPS locations at 16 KHz and 0.1 Hz, respectively. Under these settings, for a data collection round, approximately 5.46 MB have been recorded and uploaded to a web server.

Reliability: For evaluating data collection reliability, we collected data for 70 minutes during which a total of 15 surveys were delivered. The evaluation was performed inside a home using a Wi-Fi connection to upload the data. Multiple walls attenuated the Wi-Fi connection, which is realistic setup for what we expect in patient homes. Additionally, to evaluate the tolerance of AudioSense to network disconnections, we turned off the wireless connectivity of the phone 48 minutes in the experiment and restored at minute 61 of the experiment.

Figure 3 captures the reliability of the system during the 70-minute evaluation. The short red bars indicate when the data collection was initiated. As expected, consecutive bars are separated by 5 minutes, which is consistent with the experimental setup. The tall blue bars indicate the time when the data was successfully uploaded on the server. The overall reliability was 100% – all files containing the sound and GPS data have been successfully uploaded to the server.

During the first 48 minutes of the experiment, the phone had connectivity to the server. During this time interval, the average of the time from when data collection started until it was successfully uploaded was 184.48 seconds. In Figure 3, this interval is captured as the distance between consecutive short and long lines. Two factors contribute to the observed delay: a total of 180 seconds were spent collecting the data (per our setup) and the remainder of 4.48 seconds was spent upload the data. On average, the phone uploaded data at a rate of 9.756 Mbit/s.

The phone's wireless interface was turned off during the interval [48, 61] minutes. Without network connectivity, AudioSense cached data from 3 data collection rounds. Upon turning the network interface back on at minute 61, AudioSense proceeded to upload the cached files. The number on top of bar indicates the number of audio files created/uploaded within a 60 second interval. Accordingly, the number 3 on top of the penultimate tall bar indicates that the three sound files that were cached, have been uploaded within a minute.

Power consumption: The power consumption was tracked using the Power Tutor [15]. Figure 4 plots the CPU and LCD power consumption that can be attributed to AudioSense. AudioSense records data for 3 minutes during each 5 minute data collection round. This pattern is clearly visible in the figure for both the energy consumed by CPU and LCD: periods of high-energy consumption alternate with periods of no energy consumption. The LCD is used for a shorter period of time than the CPU since AudioSense starts collecting data one minute prior to delivering an alarm and turning on the LCD. During the interval [48, 61] minutes, additional energy is spent by the CPU trying to reestablish connectivity to the server. Under the considered experimental setup, the AudioSense operates at a duty cycle of 60% and the phone does not need to be recharged for at least a day.

Field Deployment: AudioSense is being used as a clinical trial that aims at evaluating the effectiveness of hearing aid technology. Currently, AudioSense has been deployed as part of three weeklong data collection sessions with 5 subjects. In contrast to the experimental setup discussed above, during the field deployment, AudioSense uploads data over the cellular network. Moreover, AudioSense operates at an 11.1% duty cycles being active (on average) for 10 minutes every 1.5 hours. Under this lower duty cycle, AudioSense can operate for three days without recharging.

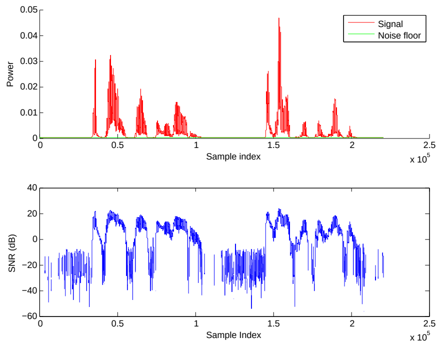


Figure 5. Inferring SNR

Computing SNR: A key factor that determines the difficulty of the listening task is the SNR. We evaluated the ability of AudioSense to compute the global SNR for noisy sound files. The noisy files were generated from a clean sound file to which Additive Gaussian Noise was added.

The SNR was estimated using the signal level and the noise floor from the power spectrum. Figure 5 plots the power of the signal and noise levels (red and green curve) for a file where the SNR was 10 db. The instantaneous SNR (computed of 0.65 ms segments) is plotted in the lower part of the graph. Figure 6 compares the actual and the estimated SNR values. On average, the SNR error was 0.62 dB but there is a clear trend of increasing error for smaller SNR values. This is expected since for low SNR values it is difficult to distinguish between signal and noise.

5 Conclusions

This paper presents AudioSense a novel system for evaluating the performance of hearing aids in the real world. AudioSense combines EMA techniques with the collection of sensor data to characterize a patient's listening context of the user. To this end, AudioSense integrates mobile phone technology with web applications. AudioSense is capable of delivering customized surveys at fixed or randomized time intervals. User feedback was integrated to refine the design of elements of user interfaces and alarms. Empirical studies show that AudioSense provided 100% reliability, supported the delivery of surveys 1.5 two hours without requiring recharging the mobile phone for two days, and provide facilities to integrate sound analysis techniques. Currently, AudioSense is used as part of a clinical trial involving the evaluation of hearing aid performance in 50 patients. The study will evaluate importance of audio samples collected in the real-world for hearing aid evaluations.

Acknowledgements

This material is based upon work supported in part by the National Science Foundation under Grant No. 1144664. Additional funding was provided by the National Institute on Deafness and Other Communication Disorders - NIH (1R03DC012551-01).

Actual (dB)	Predicted (dB)
11.52	11.59
10.56	9.76
9.54	9.76
8.56	8.54
7.56	7.750
6.59	7.17
5.57	6.01
4.56	5.11
3.60	5.17
2.56	4.29

Figure 6. SNR Estimation Accuracy

References

[1] R. M. Cox and G. C. Alexander. Maturation of hearing aid benefit: objective and subjective measurements. *Ear and Hearing*, 13(3):131–141, Jun 1992.

[2] D. S. Dalton, K. J. Cruickshanks, B. E. K. Klein, R. Klein, T. L. Wiley, and D. M. Nondahl. The impact of hearing loss on quality of life in older adults. *Gerontologist*, 43(5):661–668, Oct 2003.

[3] J. Froehlich, M. Y. Chen, S. Consolvo, B. Harrison, and J. A. Landay. MyExperience: a system for in situ tracing and capturing of user feedback on mobile phones. 2007.

[4] G. Galvez, M. B. Turbin, E. J. Thielman, J. A. Istvan, J. A. Andrews, and J. A. Henry. Feasibility of ecological momentary assessment of hearing difficulties encountered by hearing aid users. *Ear Hear*, 33(4):497–507, Jul-Aug 2012.

[5] J. A. Henry, G. Galvez, M. B. Turbin, E. J. Thielman, G. P. McMillan, and J. A. Istvan. Pilot study to evaluate ecological momentary assessment of tinnitus. *Ear Hear*, 33(2):179–290, Mar-Apr 2012.

[6] J. Ho and S. S. Intille. Using context-aware computing to reduce the perceived burden of interruptions from mobile devices. In *SIGCHI*, 2005.

[7] S. Kochkin. Customer satisfaction with hearing instruments in the digital age. *Hearing Journal*, 58(9):30–39, 2005.

[8] D. Li, I. K. Sethi, N. Dimitrova, and T. McGee. Classification of general audio data for content-based retrieval. *Pattern Recognition Letters*, 22(5):533 – 544, 2001.

[9] F. R. Lin, R. Thorpe, S. Gordon-Salant, and L. Ferrucci. Hearing loss prevalence and risk factors among older adults in the united states. *Journals of gerontology. Series A*, 66(5):582–590, May 2011.

[10] J. L. Punch, R. Robb, and A. H. Shovels. Aided listener preferences in laboratory versus real-world environments. *Ear and Hearing*, 15(1):50–61, Feb 1994.

[11] K. S. Marketrak VIII: 25-year trends in the hearing health market. *Hearing Review*, 16(11):12–31, 2009.

[12] S. Shiffman, A. A. Stone, and M. R. Hufford. Ecological momentary assessment. *Annual Review of Clinical Psychology*, 4:1–32, April 2008.

[13] Y.-H. Wu and R. A. Bentler. Impact of visual cues on directional benefit and preference: Part i–laboratory tests. *Ear and Hearing*, 31(1):22–34, Feb 2010.

[14] Y.-H. Wu and R. A. Bentler. Impact of visual cues on directional benefit and preference: Part ii–field tests. *Ear and Hearing*, 31(1):35–46, Feb 2010.

[15] L. Zhang, B. Tiwana, Z. Qian, Z. Wang, R. P. Dick, Z. M. Mao, and L. Yang. Accurate online power estimation and automatic battery behavior based power model generation for smartphones. In *CODES/ISSS '10*, 2010.