5-20-2014

Inferring Conversation Episodes From Respiratory Measurements

Md Mahbubur Rahman

INFERRING CONVERSATION EPISODES FROM RESPIRATORY MEASUREMENTS

by

Md. Mahbubur Rahman

A Thesis

Submitted in Partial Fulfillment of the

Requirements for the Degree of

Master of Science

Major: Computer Science

The University of Memphis

August 2014

ACKNOWLEDGMENTS

I would like to express my special appreciation and thanks to my advisor Dr. Santosh Kumar who has been a tremendous help to me. I would like to thank him for encouraging my research and for allowing me to grow as a research scientist. I have also learned how to communicate with other researchers, how to maintain composure under work pressure, and lots of other things that help to excel my future professional life. I would also like to thank my committee members, Dr. David Lin and Dr. Vesile Rus for serving on my committee during their busy time. I also want to mention, with gratitude, two special lab members who worked with me in this work, Amin Ahsan Ali and Dr. Kurt Plarre. A special thanks to my wife, Rummana Bari, for being an integral part of my professional and social life. I also cannot express how grateful I am to my mother, father, brothers and sisters for all of the sacrifices that they have made for me. Their prayers for me has sustained me thus far. I would also like to thank other lab members who supported me in writing and who encouraged me to strive towards my goal.

ABSTRACT

Rahman, Md Mahbubur. MS. The University of Memphis. August, 2014. Inferring Conversation Episodes From Respiratory Measurements. Major Professor: Dr. Santosh Kumar.

Automated detection of social interactions in the natural environment has resulted in promising advances in organizational behavior, consumer behavior, and behavioral health. Progress, however, has been limited since the primary means of assessing social interactions today (i.e., audio recording) has several issues in field usage such as microphone occlusion, lack of speaker specificity, and high energy drain, in addition to significant privacy concerns.

In this thesis, we present a new mobile-based system *mConverse* to infer conversation episodes from respiration measurements collected in field from an unobtrusively wearable Respiratory Inductive Plethysmograph (RIP) band worn around the user's chest. The measurements are wirelessly transmitted to a mobile phone, where they are used in a machine learning model to determine whether the wearer is speaking, listening, or quiet. This model incorporates several innovations to address issues that naturally arise in the noisy field environment such as confounding events, poor data quality due to sensor loosening and detachment, losses in the wireless channel, etc. Our basic model obtains 83% accuracy for the three class classification. We formulate a Hidden Markov Model (HMM) to further improve the accuracy to 87%. Finally, we apply our model to data collected from 22 subjects who wore the sensor for 2 full days in the field to observe conversation behavior in daily life and find that people spend 25% of their day in conversations.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

INTRODUCTION

Over the past decade, research in social signal processing [1] has demonstrated the value of automatically recording social interactions in daily life. Researchers have successfully used wearable sensors (e.g., Sociometer badge [2]) to automatically capture data on social interactions both at the individual level and at the group level, including interaction dynamics such as average duration of speaking during interactions, percentage of time speaking, etc. Via deployments of these devices on employees in different organizations such as banks, hospitals, and call centers, it has been found that such information can be remarkably valuable to the organizational management process. Using these badges in a user study showed that informal conversations account for 40-60% of the variation of the productivity of creative teams in an organization [3]. In a study on call center productivity, it was found that facilitating informal interactions led to savings of $15 million per year [4]. Researchers have also used these badges in behavioral studies to investigate the impact of social interactions on behavioral health (e.g., diet choice, exercise) [5], epidemiological behavior change [6], consumer behavior during shopping [7], etc. Such devices, however, are obtrusive in nature. For example, Sociometers are to be worn around the neck like identification badges. Though at workplaces it is not unnatural for employees to wear name tags or identification badges, the obtrusiveness of these devices make it unattractive for daily life usage.

To address the obtrusiveness issue of Sociometer like badges, state-of-the-art methods for speech detection in natural environment leverage audio signals captured on the microphone of mobile phones. They extract frequency domain features from the audio signal and can detect human voice even in the presence of ambient noise [8, 9]. This approach, however, introduces several limitations of its own. First, microphone occlusion is common in natural environments because people tend to keep mobile phones in pockets and purses. Microphone occlusion can make it difficult to collect audio signals of sufficient quality for speech detection. Second, microphone-based speech detection is not speaker-specific because the microphone can pick up the speech of anyone nearby. Thus, additional signal processing is needed to determine whether it is indeed the user's voice [10]. Third, mobile phones have computational and energy limitations. Frequent microphone sampling and feature computation in the frequency domain has a high computational and energy cost. Lastly, use of microphones also raises *privacy concerns.* A privacy study in [11] shows that 91.3% of participants were not willing to be audio recorded. Seventy-five percent remained uncomfortable even if audio is recorded only in the frequency domain. Moreover, users of audio recording devices (e.g., Personal Audio Loop) are also greatly concerned about the privacy of others (conversation partners, passers by), whose data might be captured without their consent [12].

In this thesis, as a feasible alternative, we propose *mConverse*, a mobile-phone based system that uses respiration measurements captured from an unobtrusively wearable wireless respiration sensor to infer naturally-occurring conversation events in real-time (Figure 1.1). Use of respiration measurements, eliminates the obtrusiveness, specificity, and privacy issues, inherent in an audio-based system. Wearing of respiration sensors may also provide some other benefits to the wearer. For example, psychological stress can be reliably measured from respiration measurements [13] and hence daily stress can be monitored together with conversations. Respiration measurements can also provide the intensity of physical activity, which when used with the inference of physical activity classes from the phone's accelerometers [14] can lead to accurate estimation of calorie expenditure. Similarly, respiration measurements can provide intensity of exposure to environmental pollutants, which when used with pollution exposure detection on a mobile phone (e.g., PEIR [15]), can provide an accurate estimation of the extent of daily pollution exposure.

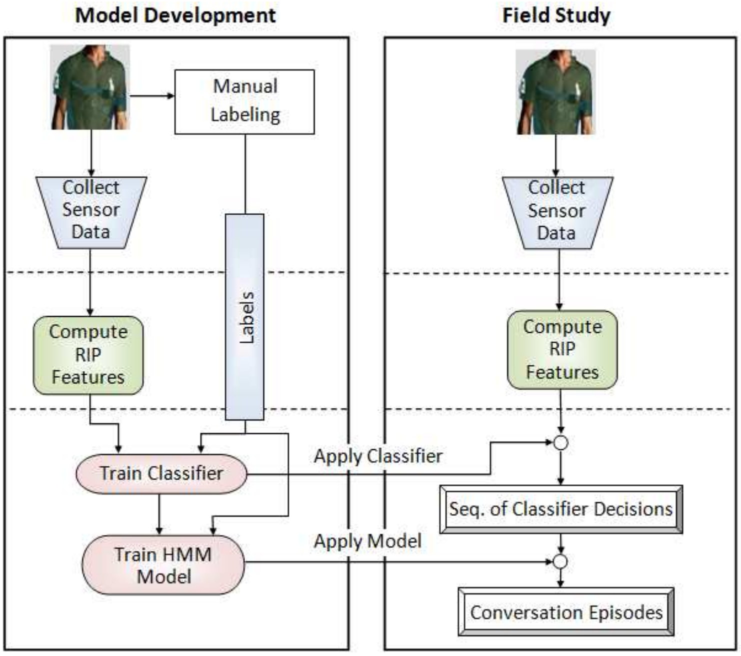


Fig. 1.1: Overview of the system: development and deployment. The speaking, listening and quiet classifier and the HMM model are developed using data collected in the natural environment (shown on the left of the figure). The classifier and the HMM model is then applied to the field data obtained from the study participants (shown on the right of the figure) to identify conversation episodes and their characteristics.

The design of our system involves several challenges. First, since the respiration sensor is a physiological sensor worn on the body, it is prone to activation by confounding factors not related to conversation. These include various forms of physical activity (e.g., walking, jogging, exercise), sneezing, yoga, etc. Second, due to movement of the body, the chest band may gradually become loose, causing a degradation in data quality, which may need to be detected and corrected quickly. Third, respiration data is transmitted to the mobile phone wirelessly and, therefore, tolerance to data lost in the wireless channel needs to be built-in to the system.

Our system carefully addresses each of these issues. First, since physical activity such as walking or running, may degrade the quality of respiration measurements due to jerks, a physical activity detection module on the mobile phone is used to infer the level of activity [13]. The inference of conversation is suspended when the wearer is detected to be undergoing significant physical activity. Second, a sensor displacement detector module on the mobile phone is used to detect when the chest band is loose [16]. Wearer of the sensor is prompted on the mobile phone to correct the placement and tighten the respiration band.

Third, in order to make robust inferences, the sensor measurements are segmented into windows so that various robust statistics can be computed over each window. From experiments conducted with data captured in realistic environments, we find that a 30 second window provides best trade-off between accuracy and robustness. With this design, each 30 second period is classified into three states — quiet, listening or speaking. In natural environment, though, the durations of speaking, listening, or quiet episodes are rarely multiples of 30 seconds. Therefore, we assign each window to a state that occurs for the maximum duration in that window. For example, if the duration of speaking is 20 seconds and that of listening is 10 seconds, in a 30 second window, then the classifier classifies the entire window as a speaking event. The windowing technique enables the system to handle the missing data issue in a hierarchical manner. Missing data within a window is handled by interpolation based methods if the amount of received data exceeds a minimum threshold, otherwise the entire window is discarded. Discarded windows produce a "missing decision" in the sequence of decisions, and are later "filled in" by a Hidden Markov Model.

To develop and validate the model, we collected 46 hours of data from 12 subjects in their natural environment that was carefully marked for beginning and end of conversation episodes, including start/end times of speaking/listening events. For this dataset, our basic model for classifying three classes (i.e., quiet, listening, and speaking) obtains 83% accuracy. We formulate a Hidden Markov Model to further improve the accuracy to 87%.

In summary, this thesis makes the following main contributions. First, we identify novel respiration features to distinguish quiet, listening and speaking states. To the best of our knowledge, this is the first work to show that inference of listening state is possible from respiration measurements. Second, we propose a data processing pipeline that can be used to make robust context inferences on mobile phones that can tolerate data lost during wireless transmission between wearable sensors and mobile phones. Third, we apply our model to data collected from the natural environment of 22 subjects to discover natural conversation behavior in daily life such as average duration and frequency of conversations, average duration of speaking and listening within conversations, and several others, as shown in Table 1.1.

Table 1.1: Summary of statistics obtained from the field data (average value and standard deviation).

Statistic	**Mean ± Standard Deviation**
Duration of conversation (min.)	03.82 ± 03.04
Time between conversations (min.)	13.38 ± 23.86
Duration of speaking (sec.)	34.20 ± 00.60
Duration of listening (sec.)	47.40 ± 06.20
Conversations per hour	02.96 ± 01.60
Percentage time in conversation	25.60 ± 05.00

Potential Applications. Automated monitoring of natural conversations and transitions between the speaking and listening states within a conversation opens the door for developing several new applications. First, a mobile phone can become conversation-aware, e.g., switch the phone to vibrate mode, update the users' social network status to "busy", etc. during conversations. Second, capturing different features of conversation, such as fraction of time spent in conversations per day, fraction of time spent speaking in a conversation, and time between successive speaking episodes during a conversation, provides useful cues to a user interested in improving his/her interpersonal communication skills. Such an application can become even more valuable if it can also record the user's stress level before, during, and after a conversation episode, using recently developed models that provide reliable inference of psychological stress using the same respiration measurements [13]. Third, linking the concomitant physiological changes and psychological stress with sensitive and specific detection of conversation will set the stage for developing more effective diagnostic and intervention methods. Such technology can, for example, help in identifying episodes of intense emotions such as anger which could precipitate cardiac events in vulnerable individuals. The technology could then be used to deliver opportunistic interventions that could be signaled to the patient in real time.

Chapter 2

RELATED WORK

There is a rapidly growing interest in inferring conversations on mobile phones so that natural social interactions can be monitored automatically. Most efforts have, however, focused on using audio recordings [8, 10, 14, 17, 18, 19, 20, 21]. Soundsense [8] was the one of the first systems to process audio signals captured on the phone's microphone to detect human speech. The energy consumption of microphone pipeline can, however, be quite high. The sampling of audio signal itself can consume energy as high as 30 times the idle state [10]. Although separate sensor boards can be attached to a phone to offload the audio sampling during quiet periods [10], it may not be convenient and acceptable to many users.For example, authors in [14] show that during classification of audio signal the CPU load rises up to 22%. In [14] the authors make necessary changes to the pipeline in order to make it less resource hungry. The authors attempt to reduce this energy requirement using stricter admission control to reduce the amount of data entering the pipeline, as well as duty cycling to reduce the audio sampling rate when the environment deemed to be silent. Moreover, the pipeline makes use of low cost classifiers and similarity measures to reduce the work load of the more computationally expensive classifiers. In spite of these advancements it is still not evident that the phone battery will last for a long enough time, which is essential in continuous sensing applications. In [17], authors try to infer conversation from audio recording using wearable microphones. In their case, multiple people have to wear recording devices each of which saves separate streams of audio features. Then they combine the streams and infer conversations. Additionally, privacy concerns arise in continuous capture of audio in the field since voices of other people may be recorded unintentionally. Several studies [11, 12] show that most participants, who are asked to carry audio recording devices, feel uncomfortable because of these privacy issues.

Some works address this issue by only capturing privacy sensitive features from audio, from which it is not possible to reconstruct the original voice [17]. Interestingly, 75% of participant in the above mentioned privacy study were still worried about privacy when they were told that the audio would be stored in the frequency domain. Use of respiration based conversation detection together with audio based systems can potentially address both of these limitations.

Processing of respiration measurements does not involve any frequency domain processing and involves computations every 30 seconds as opposed to sub-second processing. When a speaking event is detected, audio capture can be activated to obtain more specific information such as speaker identification [10]. Doing so will also significantly improve the specificity of conversation detection. Since, with this design, audio will now be captured only when the wearer is involved in conversations, privacy issues related to capture of unrelated individuals may also be mitigated to some extent.

There have been some work on the processing of respiration measurements, but the focus has largely been on health issues such as monitoring breathing disorders (e.g., sleep apnea syndrome) detection [22, 23], emotion recognition [24], and stress [13]. There have been some preliminary work on analyzing individual breath cycles during human speech [25, 26, 27, 28]. It has been observed that respiration during periods when the user is silent, is more rhythmic in contrast to periods when the user is speaking [27]. It is shown in [27, 28] that the ratio between the inhalation duration to exhalation duration is the most discriminatory feature between quiet and speaking periods. However, in these studies, respiration measurements are obtained in a lab environments and identification of conversation episode is not addressed. In a preliminary work [29], we showed that the ratio between inhalation and exhalation durations as a single feature is not robust enough for discriminating speech and quiet when respiration measurements are collected in the field.

In summary, none of the above mentioned works show how listening can be detected. Detection of listening is necessary to reliably mark the entire conversation episode. This thesis is the first to propose a set of features computed from the respiration signal, and a robust classifier to distinguish among quiet, listening, and speaking events.

Chapter 3

DATA COLLECTION

In this chapter, we describe the sensor suite we used to capture respiration measurements and the data collection experiment for collecting respiration measurements that we use for developing classifiers.

3.1 The Sensor Suite

We used the AutoSense sensor suite [30] that includes a Respiratory Inductive Plethysmograph (RIP) band to measure relative lung volume and breathing rate (see Figure 3.1). RIP uses a conductive thread that is sewn in a zigzag fashion to the elastic band. An alternating current source is applied to the resulting loop of wire, which, in turn, generates a magnetic field that opposes the current whose strength is proportional to the area enclosed by the wire according to Lenz's law. The ratio of the magnetic flux to the current is called self-inductance. Therefore, changes to the chest circumference can be measured by measuring the changes to the self inductance of the band. The inductance measurement purely depends on the geometry of the band and is not related to the tension in the band. As a result, the measurement is not prone to the trapping of the band and associated artifacts due to changes in tension. The sensor suite also includes a 3-axis accelerometer which is used to detect the level of physical activity of the wearer. The sensors transmit data to an Android mobile phone in real-time over a low-power wireless link.

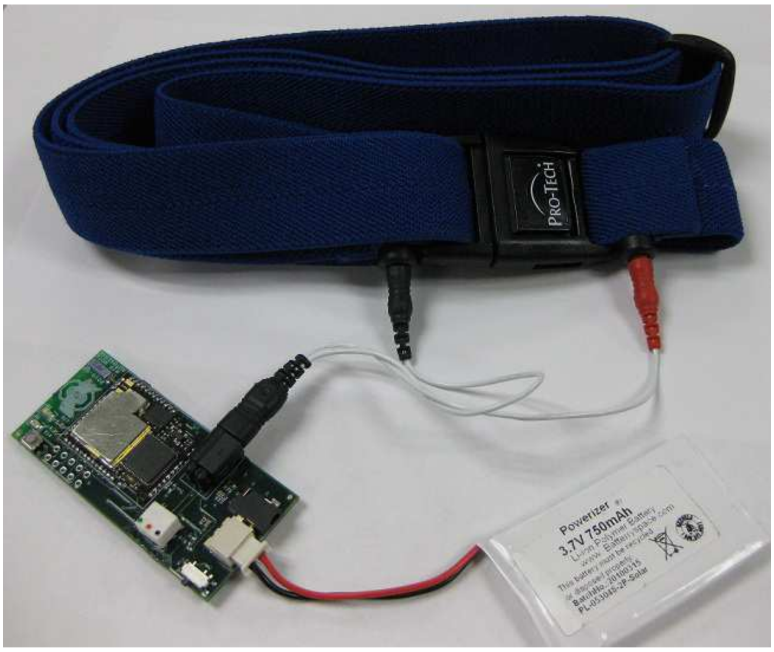
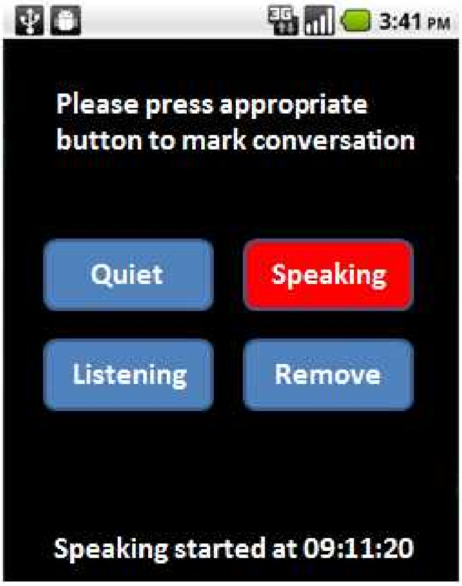


Fig. 3.1: Respiratory Inductive Plethysmograph (RIP) chest band (in blue color) and an uncovered AutoSense sensor board.

3.2 Data Collection for Model Development

3.2.1 Data labeling application

We have developed an android application on top of FieldStream [31] framework to collect data, and associated ground truth labels. The application has four buttons for data labeling - “Speaking”, “Listening”, “Quiet”, and “Remove”. It saves ground truth labels to a separate text file other than the data file. Figure 3.2 shows the interface of the application. Upon every button press (e.g., speaking), it saves the timestamp of the moment as the start of the current state. When there is a change in state, the labeler (who is labeling carefully) presses the button associated with the new state. Start of the new state indicates the end of the previous state. However it is very challenging to capture ground truth of each state during ongoing conversation. If the labeler wants to remove some erroneous label, they can do it by pressing the “Remove” button. It removes the last entry from the label file. At the end of the interface, the application shows the start time of the



Please press appropriate button to mark conversation

Speaking started at 09:11:20

Fig. 3.2: Data labeling application on android smartphone. The data labeler marks the current state by pressing corresponding button the screen. “Remove” button helps to remove erroneous label. At the end of the interface, there is a text to show the current label with corresponding start time.

current state so that the labeler can be aware of the current ongoing state, and ongoing ground truth label.

3.2.2 Data collection

We collected data from 12 subjects (10 men, 2 women) for a total of 2,772 minutes (46.2 hours) of natural conversation. By natural conversation, we mean unscripted naturally occurring conversation in our daily life. The subjects wore the chest band in their natural environment and were accompanied by an observer. The observer marked the start and end times of the wearer’s speaking, listening, and quiet episodes on the mobile phone that received the respiration measurements via wireless channel. After extracting the data, we assign labels to each 30 second window. A window is assigned to a label if corresponding event occurs for $\geq 66\%$ of the total duration of the window. Otherwise, we label the window as lost data.

Chapter 4

DISTINGUISHING QUIET, LISTENING, AND SPEAKING

We describe the development and evaluation of our model that classifies 30 second windows of respiration measurements in quiet, listening, and speaking events. Figure 4.2 shows a snapshot of respiration signals during listening, quiet, and speaking segments to illustrate the difference in breathing pattern during these three states. Figure 4.1 shows the entire procedure for identifying conversation episodes from respiration measurements. This sections describes the data processing pipeline stages for preprocessing, feature computation, and classification. Hidden Markov Model and postprocessing, which are applied to the output of the classifier to further improve its accuracy, is described in Chapter 6.

4.1 Preprocessing & Identification of Respiration Cycles

Before computing features, we segment respiration measurements in 30-second intervals, identify windows with sufficiently valid data (i.e., admission control), impute missing samples, and remove outliers. These measurements are then used to identify the locations of the peaks and valleys of each respiration cycle. These steps are illustrated in Figure 4.3 and described in the following.

4.1.1 Raw Data Segmentation

We could follow two approaches for windowing – overlapping windows and non-overlapping windows. Overlapping windows are able to capture the characteristics signature of respiration more precisely, but are too intense for implementation on resource-constrained mobile phones that may be making multiple concurrent rich inferences such as stress, posture, physical activity, location, etc. In a 30 second window, we usually observe, on average, 5-10 breathing cycles. This number is large enough to allow us to compute various statistics over feature values. Figure 4.4 shows how classification accuracies vary with different window lengths. Although longer windows give slightly better accuracy, it reduces

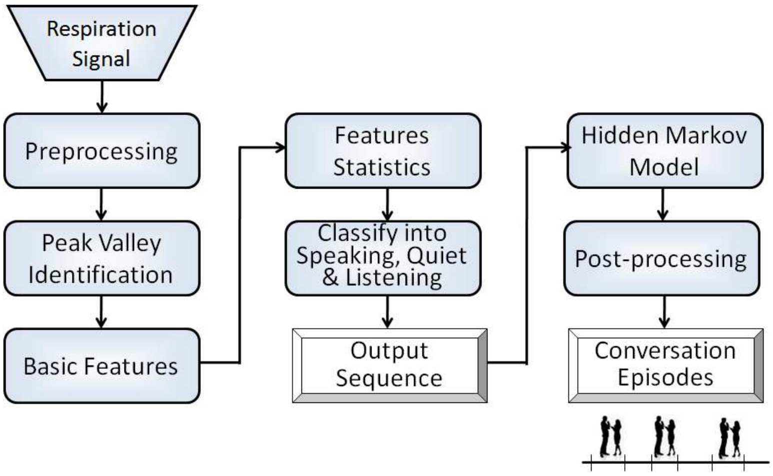


Fig. 4.1: Processing pipeline of the conversation detection system.

the granularity of measurement since one window can only be assigned to one class of event (i.e., speaking, listening, or quiet). Since we find that increasing the window length to > 30 seconds provides only marginal improvement in accuracy, we use 30 seconds for the window length.

4.1.2 Admission Control

We use admission control to decide which windows to pass through to upper layers since a window may not have enough data of good quality to compute feature statistics with sufficient reliability. We note that data quality may be affected by noise, sensor displacement, and sensor detachment. We use three criteria in admission control.

Impact of Physical Activity

Physical activity affects the breathing pattern, making it harder to reliably infer speaking events when the wearer is undergoing physical activity. Physical activity is detected automatically from accelerometer sensors in the AutoSense sensor suite. All windows affected by physical activity are ignored and considered lost.

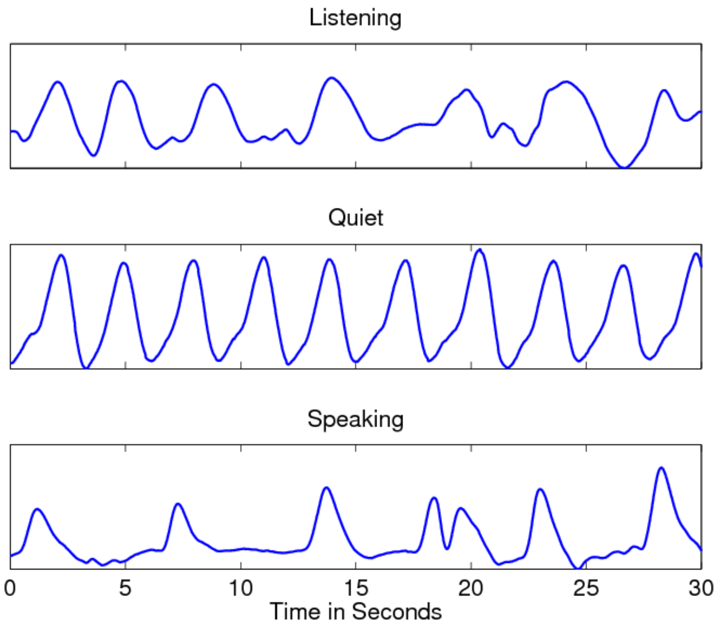


Fig. 4.2: Representative respiration signal during listening, quiet and speaking events. Y-axis represents ADC values which corresponds to the amplitude of relative lung volume.

Impact of Data Quality

In order to infer the context with higher confidence from sensor data, we must ensure good quality of measurements. Several events adversely affect the quality of data, such as loosening of the RIP band, sensor displacement, and sensor detachment. These events are detected by the data quality detector [16], and all windows affected by them are ignored.

Handling Lost Data

Some measurement samples are lost in the wireless channel. Figure 5.1 shows the cumulative distribution of data lost in 30-second windows that were obtained from 22 subjects who they wore the RIP sensor for 2 days in the field. We find that 87% windows had a loss rate of $< 34\%$. Hence, we use this as the threshold, i.e., all 30-seconds that have $\geq 66\%$ of samples are accepted.

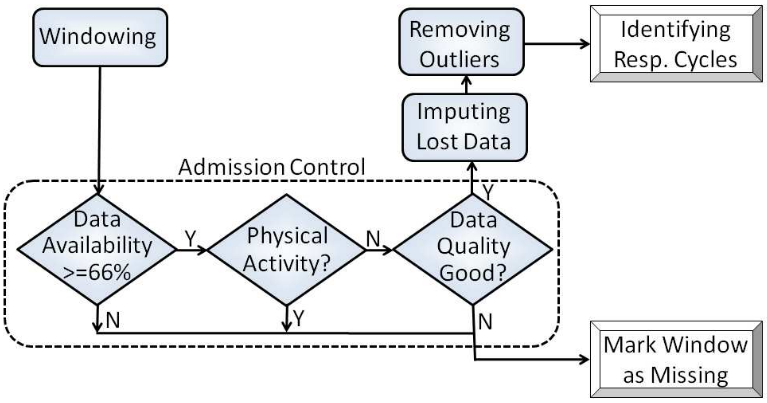


Fig. 4.3: Preprocessing of the respiration signal. Figure shows the flow of windowed raw respiration signal. Windows of respiration signal are filtered through the admission control. There are three types of admission control - checking data availability, presence of physical activity, and checking quality of signal. When a window of signal fails to pass the admission control, we mark the window as missing data. Otherwise, we interpolate, and remove outliers (e.g., spikes) from the raw signal, and finally, apply our peak-valley detector to extract time domain features.

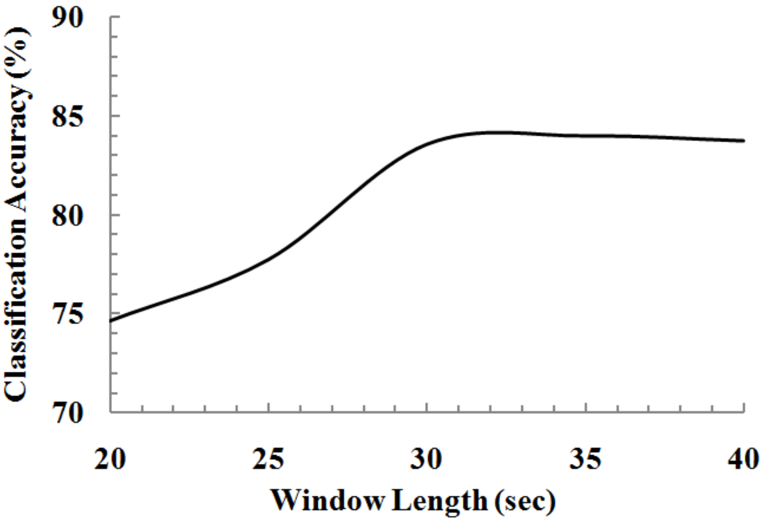


Fig. 4.4: Window length (sec) vs. accuracy(%). It shows how classification accuracy (speaking, listening, and quiet) changes with the change of the window size. Based on the experiment, we take 30 seconds window where accuracy is around 85%.

4.1.3 Imputing Lost Data

In windows of measurements that have $\geq 66\%$ of valid samples, we apply spline interpolation to fill in the missing values. We use the standard spline function from MATLAB.

4.1.4 Outlier Removal

There are several methods to detect outliers from data [32]. Since quartiles are less sensitive to spikes that may appear in respiration measurements collected in noisy field environments, we use them for outlier detection. We first find the upper quartile (LQ) and lower quartile (UQ) values in a window. The interquartile range (IQR) is the difference between the upper and lower quartiles, i.e., $IQR = UQ - LQ$. Outliers are defined as those points that are $\geq 1.5IQR + UQ$ or $\leq LQ - 1.5IQR$.

4.1.5 Identification of Respiration Cycles

To identify a respiration cycle, its peaks and valleys are to be located. We use a modified version of the peak-valley detection algorithm presented in [25, 26] to locate the peaks and valleys for each respiration cycle in a 30-second window, after applying the above described preprocessing steps. In order to remove spurious peaks, we set a lower threshold for a measurement to be considered a peak. From experiments, we find that setting this threshold to the 75th percentile of the signal amplitudes for each window works well. We also require that duration between two successive peaks must be at least 1.5 seconds. It means that duration of each respiration cycle can not be as short as 1.5 seconds for an individual. Through visual inspection, we find that the performance of the peak-valley detection algorithm has 96.11% accuracy. If the number of valid peaks and valleys in a segment is sufficient (more than 3 respiration cycles) then we calculate features from this window. Otherwise, this window is ignored.

4.2 Feature Identification

We identify six distinct features that are computed from the respiration signal. We identify three features from existing work and propose three new features. We investigated several features from visual inspection and narrowed it down to three after determining their discriminatory power using feature selection algorithms. Computation of the features involves the identification of the respiration cycles, which are composed of an inhalation and an exhalation period. We now define these six features in the following and illustrate them in Figure 4.5.

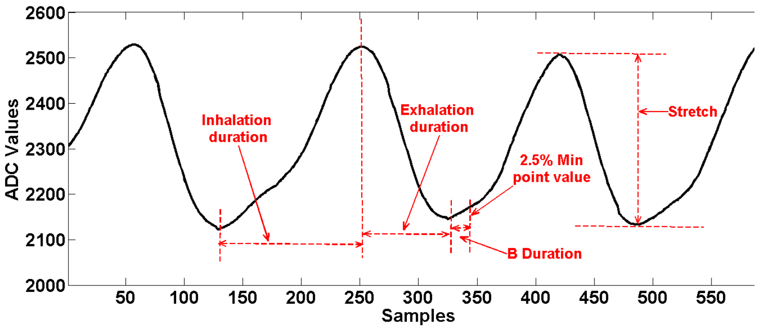


Fig. 4.5: Illustration of features. All of them are time domain features, and are based on peaks and troughs.

4.2.1 Existing Features

We first describe three features that have previously been proposed for identifying speaking events from respiration [27]. **IE ratio** is defined as the ratio of inhalation duration to the exhalation duration of a respiration cycle. IE ratio has been traditionally referred to be the most distinguishing feature for this classification. **Inhalation duration** corresponds to the time elapsed from a valley of a respiration signal, to the subsequent peak. The amplitude difference in signal values between these points is the maximum expansion of the chest during a respiration cycle (see Figure 4.5); **exhalation duration** corresponds to the time duration between a peak and the subsequent valley. Exhalation duration during speech tends to be longer than that of silence. Use of only IE ratio did not prove to be sufficient in natural environment; it provides a classification accuracy of only 64.12% for speaking vs. not-speaking events.

4.2.2 New Features

We now describe the three new features we identified in this work. During this feature identification, our lab member Amin Ahsan Ali worked with me. We observed that people tend to hold their breath while speaking, pause to take a deep breath, and repeat this cycle. The features proposed below are meant to capture this distinguishing pattern.

- **First Difference of Exhalation** is derived by computing the first order differences of the exhalation durations. This difference is observed to be lower during quiet events because the breathing pattern is regular.
- **Stretch** is the difference between the amplitude of the peak, and the minimum amplitude the signal attains within a respiration cycle (see Figure 4.5). Stretch values are found to be large during speaking events as people tend to take relatively deep breath as they talk.
- **Pause Duration (B-Duration)** is defined as the time the signal continues to stay within 2.5% of the minimum amplitude. It is found to be longer during speaking events because we tend to hold our breath during speaking.

4.2.3 Feature Extraction

There are multiple respiratory cycles in a 30-second window and each cycle produces a value for each of the 6 features. To reduce the effect of noise and outliers (e.g. spikes in the respiration signal due to movement) we compute four statistics over the values of each feature produced in a 30-second window. We find mean, median, standard deviation and 80^{th} percentile to be useful features for our classification. Thus, we consider a total of 24 features in training the classifiers. As an illustration, Figure 4.6 shows the 80^{th} percentile values of stretch (computed for each window) in different user states. It can be observed that these values are visibly higher during public speaking and non-public speaking events whereas they are the lowest during quiet periods.

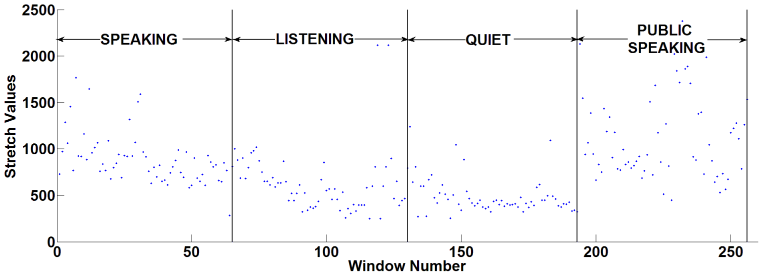


Fig. 4.6: 80th percentile of Stretch values for each window in different user states.

4.2.4 Normalization

We observed that the variability of the respiration signal across subjects is quite high. Therefore, we normalize them to reduce these inter-subject differences for building a robust classifier. In order to normalize the features, we first compute each feature. Then, we compute mean and standard deviation for each feature, for each subject, across all the events (i.e., quiet, speaking, listening) for that subject. This corresponds to the global mean and standard deviation for the specific subject. We then compute the z-score of the features for each window, by subtracting the mean and dividing by the standard deviation. The z-scores of the features are then used for classification purposes.

4.2.5 Feature Selection

Feature selection methods typically fall into two broad categories: (a) *wrappers*, which evaluate the set of features using the classification algorithm that is to be applied to the data, and (b) *filters*, which are independent of the classification algorithm and evaluate the set of features by using heuristics based on general characteristics of the data [33]. Wrappers are less general in the sense that the feature selection process, in the case of wrappers, is tightly coupled with a classification algorithm, and must be re-run when switching from one classification algorithm to another.

We use the filter approach for feature selection proposed in [33]. We employ the filtering method called CFS (Correlation based Feature Selection), which removes irrelevant and redundant features to output a feature subset that contains features highly correlated with (i.e., predictive of) the class, yet uncorrelated with (i.e., not predictive of) each other. An important criterion of filter based methods is the direction of search in the feature space. The best fit method used with CFS algorithm searches the space of feature subsets by greedy hill-climbing augmented with backtracking. Setting a bound on the number of consecutive non-improving nodes permitted controls the level of backtracking, i.e. limiting the number of fully expanded subsets that result in no improvement. We set the stopping criterion to five and use Best First Method with forward, backward, and bi-directional searches.

Upon executing feature selection, we find that at least one of the statistics computed over each of the six features are selected for classification (see Table 4.1). We use these selected feature statistics to train the classifiers.

Table 4.1: Feature selection based on the CFS algorithm with best fit search.

Basic Feature	**Feature Statistics**
Inhalation Duration	Standard Deviation
Exhalation Duration	Mean, Median, 80^{th} Percentile
IE ratio	Mean, Median, Standard Deviation, 80^{th} Percentile
Stretch	Mean, Median, 80^{th} Percentile
Pause Duration	Median, Standard Deviation, 80^{th} Percentile
Exhal First Difference	Mean

Chapter 5

TRAINING AND CLASSIFICATION

We consider three challenges during classification. First, how to label each window during training, given that each 30 sec window can have multiple transitions of speaking, listening, and quiet? Second, how to determine the tolerance level for data loss in each window? Third, how to identify listening from respiration measurements, which has not been attempted yet?

For the first case, we define purity of the ground truth. If we have the entire 30 sec period of speaking, quiet, or listening, then the window has 100% pure label. If the window has more than one label, then the window will have impurity in its label. We consider the label that occurs for the longest duration in the window as its ground truth. We do experiment training on 100% pure labeled data and test that model with 100% pure labeled data, and impure labeled data. We also try training on impure labeled data and test on 100% pure, and impure labeled data. We did not find significant difference in accuracy. However, training on impure labeled data and testing on impure label data produces better results. In fact, this condition matches with most of the typical conversations in real life because speaking and listening switches rapidly within each 30 sec time span.

For the second challenge, we calculate the probability of data missing in the field from our field study. We show the cumulative density function of missing data in the figure 5.1. Since 87% of the windows have $<$ 34% missing data, we train our classifier with tolerance of 34% missing data rate.

5.1 Identifying Listening Events

To the best of our knowledge, there is no classifier that distinguishes listening from speaking and quiet states, using respiration measurements. To detect listening from respiratory measurement is especially challenging, since, depending on the duration of listening, the signal can be similar to quiet or similar to speaking.

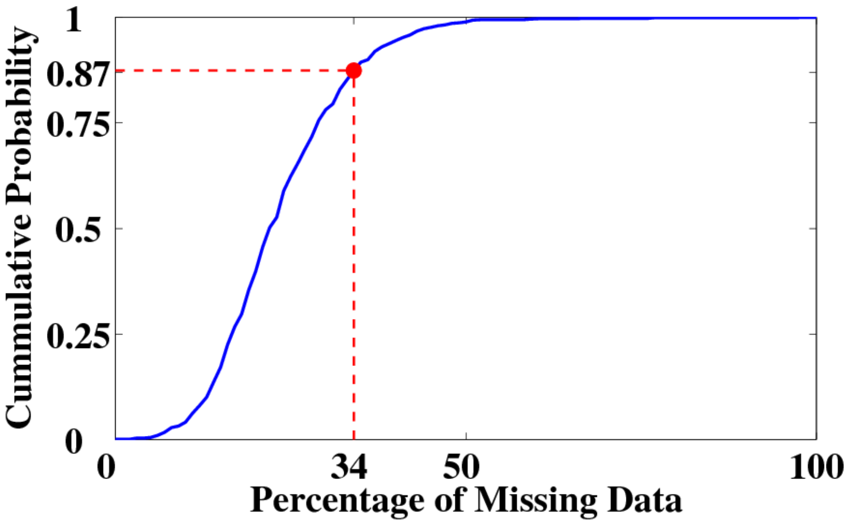


Fig. 5.1: CDF of missing respiration data in field. The figure shows that more than 87% data has missing rate <= 34%.

In a two party conversation, listening contains some non-verbal expressions to demonstrate engagement, which produces some irregularity in the respiration cycle compared to the quiet state. It needs more air to the lung. Stretch of each cycle, therefore, falls in between the stretch of the speaking and quiet respiration cycles, and becomes a distinguishing feature for identifying listening events.

5.2 Classification Algorithm & Accuracy

We train Decision Tree (J48), SVM, and AdaBoostM1(J48) [34] classifiers using both the selected feature set and the entire feature sets using Weka Tool [34, 35] on the 12-subject data set described in Section 3.2. We use 10-fold cross validation to obtain classification accuracies. The results appear in table 5.1. They correspond to training and testing with impure labeling. Validation using

Table 5.1: Comparison of classifiers for impure training and impure testing.

Classifier	All Features				Selected Features			
	Accuracy	Kappa	Precision	Recall	Accuracy	Kappa	Precision	Recall
J48	0.79	0.68	0.82	0.81	0.77	0.67	0.83	0.76
Adaboost (J48)	0.84	0.75	0.88	0.82	0.83	0.74	0.88	0.80
SVM	0.83	0.74	0.87	0.81	0.84	0.76	0.87	0.82

66-34% split also produces similar classification accuracy. The best accuracy (83.51%) is found using the boosted decision tree, although the performance of other classifiers are comparable. Also, there is no significant change in the classifiers' performance when we use only the features selected by the CFS algorithm. For ease of implementation, we choose the decision tree model for mobile phone implementation.

We find that listening is sometimes misclassified as speaking. This is not an issue though for identifying conversation episodes since conversation consists of both speaking and listening. Since quiet states mark the boundary of a conversation, misclassifying listening as quiet is more problematic. This misclassification, however, is limited to 6.25%. The classification accuracy is further reduced by using a Hidden Markov Model (HMM) to find the most probable sequence of speaking, listening, and quiet, to correct the results of the classifier, as discussed in Chapter 6.

Chapter 6

CONSTRUCTING CONVERSATION EPISODES

We define a conversation episode as the period between two successive quiet events. Because each classification window length is 30 seconds, the shortest duration of a conversation that can be identified using the method described in this work is 30 seconds. A conversation episode can be constructed directly from the output of the classifier (in Chapter 5) by locating sequence of quite states that demarcate the boundaries of a conversation. The accuracy of the classifier, therefore, directly impacts the accuracy of constructing conversation episodes. We use two methods to improve the accuracy of classification, which will also improve the accuracy of conversation characterization. We develop a Hidden Markov Model (HMM) to leverage the fact that we usually transition among speaking and listening states during a conversation, and a conversation event is preceded and succeeded by a sequence of quiet states. We then apply some postprocessing rules to the output of the HMM.

6.1 HMM for Conversation Episode Identification

An HMM is a Markov process comprising of a set of hidden states and a set of observables. Every state may emit an observable with a known conditional probability distribution called the *emission probability.* Transitions among the hidden states are governed by a different set of probabilities called *transition probabilities.* Upon executing on a sequence of hidden states, an HMM produces a sequence of observables as its output. While the output (i.e., the list of observables) can be observed directly, the sequence of hidden states that were traversed in producing the observed output is unknown; the problem is to determine the most likely sequence of states that may have produced the observed output. Viterbi decoding [36] is a dynamic programming technique to find the maximum likelihood sequence of hidden states given a set of observables, emission probability distribution, and transition probabilities.

In our HMM model, there are three states — *quiet-state*, *speaking-state* and *listening-state* and three observables — *quiet-respiration*, *speaking-respiration* and *listening-respiration.* We introduce a new observation to indicate missing data, which we call *MissingObservation.* Each state emits observables depending on the emission probability. The system can start from any of the three states. It may be natural to assume the initial state to be the *quiet-state*, because in most cases, the person will be quiet while being fitted with the chest band. But, the data collection does not start immediately after a person puts on the sensors. It might start later when the mConverse system on the mobile is activated, at which time the wearer may be engaged in a conversation. Moreover, estimating such probabilities is hard in practice, and therefore, we use an uninformative prior, i.e., the initial probabilities for all the three states are equal to 1/3.

6.1.1 Emission Probabilities

We have four observations — *speaking*, *listening*, *quiet*, and *missing data.* If an observation is available, we calculate the emission probabilities from the confusion matrix (6.1) we get after applying the classifier described in Chapter 5, and set the emission probability of *MissingObservation* to 0 from all states. For example, the quiet state can be detected with 81.71% accuracy by the classifier, and, therefore, the emission probability from *quiet-state* to *quiet-respiration* is 0.82. If an observation is missing, then the emission probability of *MissingObservation* from any state is set to 1, and the emission probability of other three observables from any state is set to 0. In other words, the Viterbi algorithm is guided to retain the current state if an observation is missing.

Table 6.1: Confusion matrix for Speaking, Listening and Quiet classification after applying the AdaboostM1(with J48) classifier.

Classified as ⟶	**Quiet**	**Listening**	**Speaking**
Quiet	**0.82**	0.12	0.06
Listening	0.06	**0.81**	0.13
Speaking	0.05	0.08	**0.88**

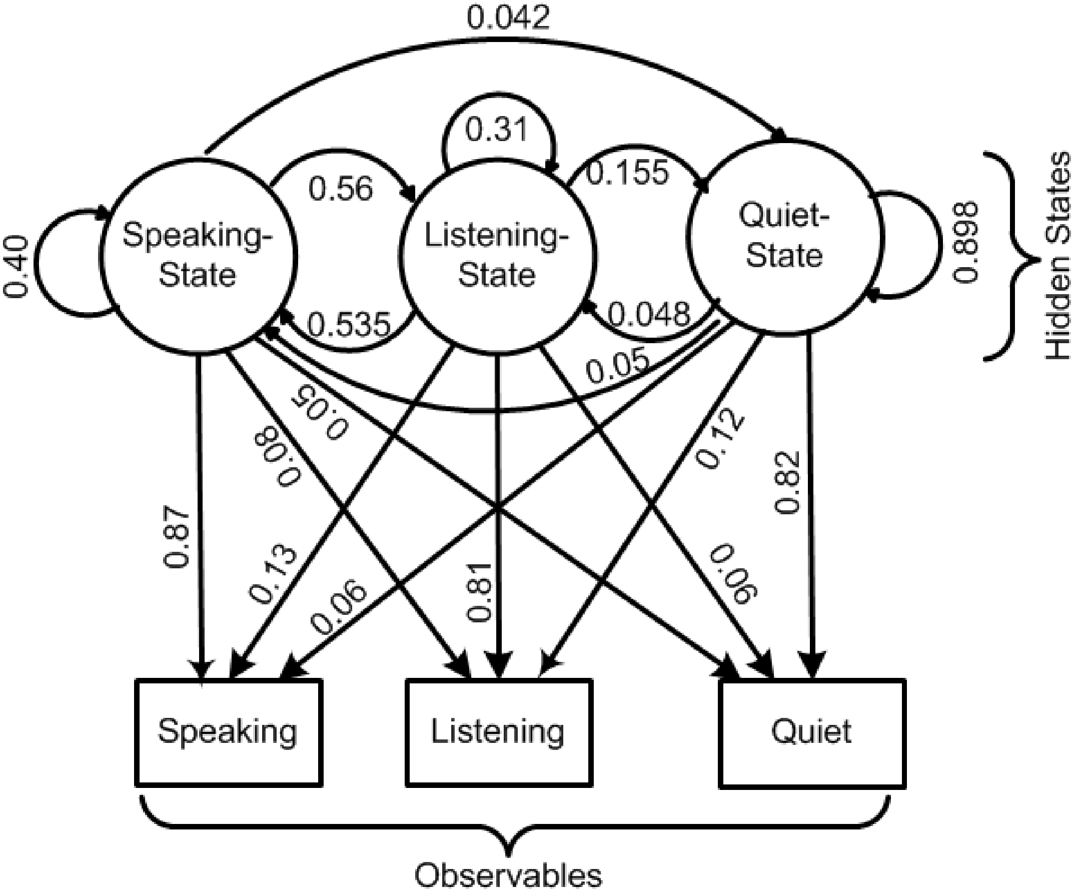


Fig. 6.1: HMM state description for conversation episode identification. Three circles indicate three hidden states. Three rectangles indicate three observables. All the emission probability are shown with the outgoing arrow from state to the observables. Arrows from state to state indicate transition probabilities.

6.1.2 Transition probabilities

We calculate transition probabilities from the ground truth labeling of various conversations while varying the subjects and their contexts. From the conversation data, we calculate the transition probabilities by considering the ratio between the frequency of transitions from an origin state to a target state, to the frequency of the origin state in the data set. For example, we calculate the transition probability from speaking to quiet state as the ratio of the frequency of transitions from speaking to quiet state, and the frequency of the speaking state; a majority of speaking states may transition to listening state.

We train our HMM from carefully labeled (by an observer) conversation data from 12 subjects in the natural environment as described in Section 3.2. We calculate transition probabilities based on the labeled training data. We compare the current state with the previous one and count the window-wise transitions. In our case, we have three states and nine transition probabilities. See Figure 6.1 for the specific numbers we find from our data set for transition and emission probabilities.

The 30-second windows of input respiration measurements are classified as a sequence of speaking, listening and quiet events from the classifier described in Chapter 5. This sequence is then fed to our HMM which generates smoothed sequence as its output of those three events. Although Viterbi algorithm can be applied to the entire stream at a time, if processing the respiration measurements offline, in this work, we apply the HMM to segments of 10 windows. We thus obtain revised assignment of each window into speaking, listening, or quiet, including those that may be missing. If a majority of windows in a block of 10 windows are missing, then the missing windows are not assigned any event label.

6.1.3 Post-Processing

We use two post processing steps to the output sequence produced by HMM to further improve the classification accuracy. First, if we find only some listening events inside a long sequence of quiet, then we convert it to a quiet event. Since these listening events are not backed up by speaking events, they are unlikely to be part of a conversation.

Second, we apply outlier removal algorithm (described in Section 4.1) to the total duration of speaking, listening and quiet sequence. This procedure removes unusually long or short sequence of speaking or listening events since in a typical conversation, speaking or listening can not be unusually too long or too short.

6.1.4 Evaluation

As described earlier, a 30 second window may have a mix of three possible events — speaking, listening and quiet. This introduces impurity in ground truth labeling. We find that training the classifier using data with impure ground truth labeling and testing on impure data gives the best results among all the cases. We get 83.51% classifier accuracy for three classes after applying the Adaboost classifier described in Chapter 5 (see Table 5.1). After applying HMM and postprocessing, it improves to 86.74%. Moreover, when we consider a binary class (conversation or not) classifier, we achieve more than 90% accuracy. The final confusion matrix is shown in Table 6.2.

Table 6.2: Confusion matrix after postprocessing.

Classified as ⟶	**Quiet**	**Listening**	**Speaking**
Quiet	**0.90**	0.07	0.04
Listening	0.12	**0.83**	0.05
Speaking	0.02	0.08	**0.89**

Chapter 7

CONVERSATION BEHAVIOR IN REAL-LIFE

We applied our model (developed from labeled training data) to the respiration data collected from 22 participants (11 men, 11 women) without ground truth, who wore the sensors for awake periods (12-14 hours) on two non-consecutive days in their natural environment[1] to derive interesting observations of conversation behavior in daily life. We note that since the participants were all students at University of Minnesota, Duluth, these observations may be limited to the conversation behavior of students. However, since several user studies involve students as subjects, these observations may inform scientific studies of social interactions, social support, and conversation behavior among students.

7.1 Results

We computed the speaking, listening, and quiet states. These were then used to compose conversation episodes and period between successive conversations. Figure 7.1 shows the average duration of conversation and average time between successive conversations for each subject, labeled by gender to observe any gender bias. We notice that the average duration of conversation is limited to 5 minutes, while average time between conversations is limited to 30 minutes. Note that these are averages and may not represent individual episodes.

Table 1.1 summarizes mean values of various measures of interest (e.g., conversation duration, speaking duration, frequency of conversations, etc.). We observe that conversations are, on average, short and frequent, with an average frequency of 2.96 conversations per hour. This is not unexpected, as all participants were university students, and the data included in the analysis was collected during weekdays, between 8am and 10pm. During this time, students talk mostly to their peers, between classes, for example, to discuss material, or compare homework

[1] For details of the study design and protocol, we refer the reader to [13].

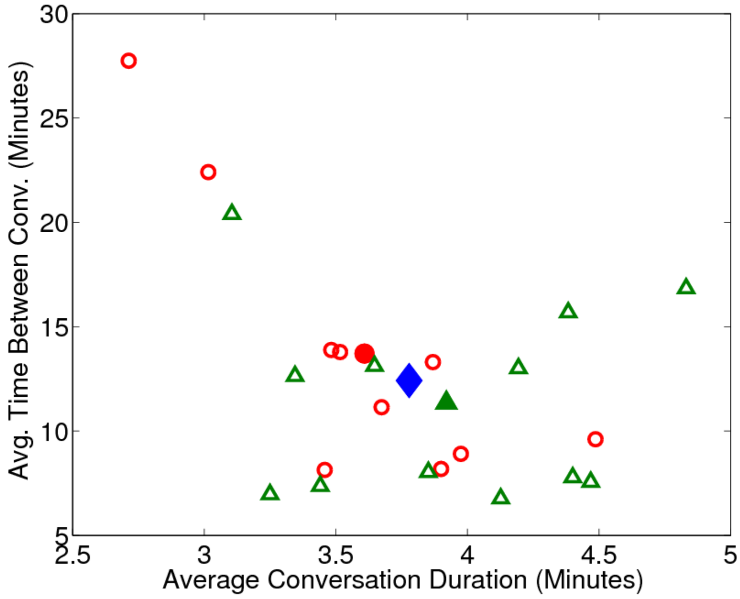


Fig. 7.1: Average (over both days) of duration of conversation and between conversation (quiet) times, for each participant. Male participants are represented with circles, females with triangles. The average times for males and females are shown with a large circle and triangle, respectively. The total average is shown as a diamond.

results. We also observe that listening segments are, on average, longer than speaking segments. In a two party conversation, we expect each person to speak for about half of the time (if we average over all conversations and participants). If we also consider group conversations, then listening segments should be longer than speaking segments. Thus, on average, we expect longer listening times than speaking. Notice also that, due to the way in which we discretize time, we cannot measure times shorter than 30 seconds. We leave for future work, to find ways of measuring shorter intervals.

We notice that the average duration of conversations per hour is about 25.6*60=18.6 minutes. This is in contrast with the results obtained in [37] in which the average time the energy on the microphone was above a given threshold was found to be close to 30 minutes per hour. This difference might be explained by the fact that the population included in the study in [37] consisted of students, faculty, and staff members from the MIT Media Lab, the duration of observation was mostly work hour (i.e., 10-5), and students were members of research groups, while the population considered in our study consisted of only undergraduate students, and covered their entire awake hours.

Figure 7.2 shows the frequency of conversation of various durations across all 22 participants. We observe that conversations of shorter durations are more frequent, with length 2 to 4 minutes being most dominant.

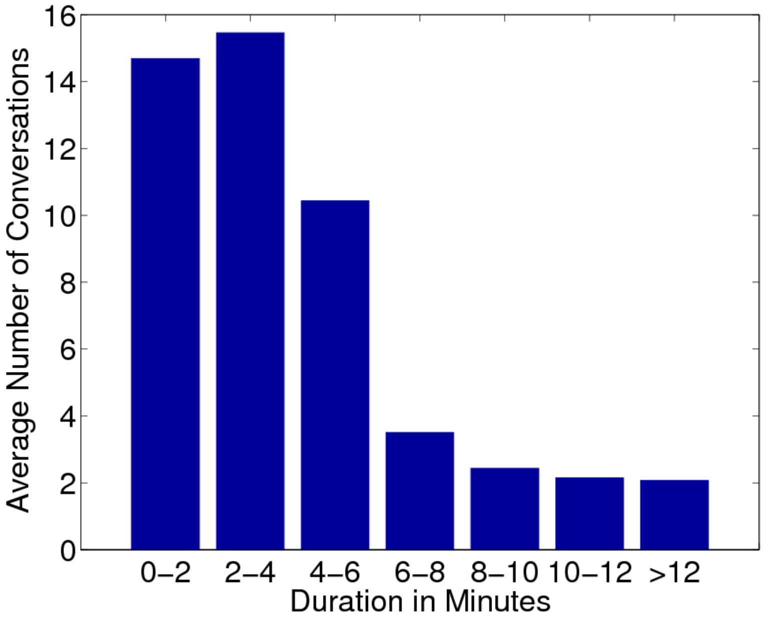


Fig. 7.2: Average number of conversations of a given duration, per person, per day.

Chapter 8

MOBILE SYSTEM IMPLIMENTATION

We have implemented our system using the FieldStream [31] framework on an Android smartphone. When the user wear the respiration chest band, and our application can wirelessly receive respiration data on the phone. We compute the selected features from the respiration signal, as described above. These features are feed into the decision tree (J48) classifier to get inferences of states inside conversation episodes. We selected decision trees since they provide high accuracy at a low computational cost compared to other machine learning algorithms.

8.1 Smartphone Application

Figure 8.1 shows a prototype of a binary version of the application which visualizes conversation states in real-time. It has two states – speaking, and not speaking (quiet). It shows the current state in red, and previous state in blue. When the user in a conversation, there will be switching between speaking, and not-speaking based on the state of the user in a conversation. Moreover in back-end, it saves conversation logs to a file to analyze or visualize the history of conversation in offline. These offline analysis and visualization can help people to become aware of their contribution in conversations. If the stress information is available associated with each conversation episode, it can help to figure out stressful conversations for the user.

One of the important feature of this conversation monitoring system is that it is very specific to the user, and logs conversation data in very privacy preserving manner. Since the conversation log is very specific the user, it may not be create privacy concern to the conversation partner as opposed to the microphone based conversation detection system.

8.2 Energy Consumption

We run our respiration based speaking detection application on Android G1 smartphone, and log the battery lifetime. To compare with microphone based conversation detection [8], we also run the microphone audio logging continuously on the phone with 8kHz sampling rate. From figure 8.2, we see that the microphone based version consumes significantly higher amount of battery than the respiration based system.

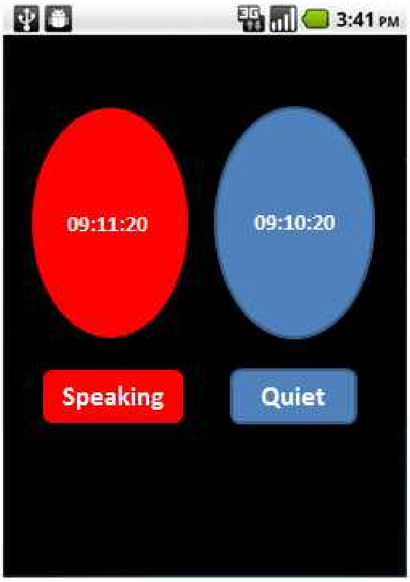


Fig. 8.1: Mobile System Implementation. The phone wireless collect respiration data wirelessly from the RIP band, and classify whether the current state is speaking.

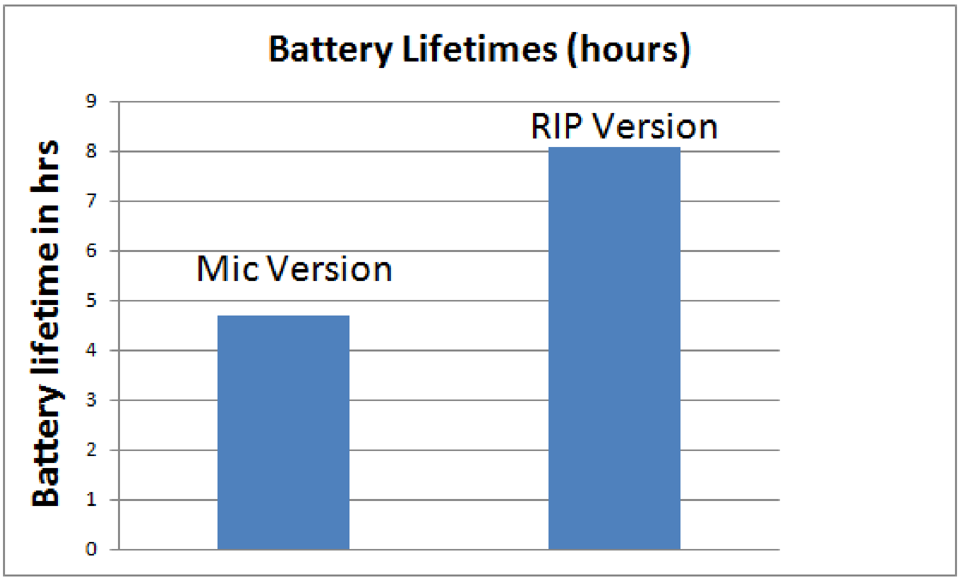


Fig. 8.2: We compare the energy consumption between microphone based conversation capture, and respiration based conversation capture. The battery consumption by the microphone based system is significantly higher than respiration based conversation detection on android smartphone.

Chapter 9

CONCLUSION AND FUTURE WORK

In this thesis, we presented *mConverse*, a respiration event classification system specifically designed for resource limited mobile phones. In contrast to traditional audio context recognition systems that are offline, *mConverse* performs online classification at a lower computational cost but yields results that are comparable to offline systems. We introduce conversation episode identification from respiration signal of a human subject by classifying them into speaking, listening and quiet events. For these classification, we propose several new time domain features from respiration which are different from the traditional features and light-weight from computational requirement. Using *mConverse* in real-life can help enhance the scientific studies of social interactions and help individuals reflect upon and improve their social interactions. Its usage together with processing of audio data captured on the microphone can help further characterize the content of conversation (e.g., how frequently the wearer discusses health issues in chronic care conditions). Capture of audio data in field can also help further validate the model proposed in this work.

This work opens up several new opportunities for future work. For example, automatic detection of other daily behaviors of interest such as laughing, singing, eating, drinking, etc., from respiration measurements on the mobile phone can be investigated.

REFERENCES

[1] A. Vinciarelli, M. Pantic, H. Bourlard, and A. Pentland, "Social signal processing: state-of-the-art and future perspectives of an emerging domain," in *ACM international conference on Multimedia*, 2008, pp. 1061–1070.

[2] D. Olguín and A. Pentland, "Social sensors for automatic data collection," in *Americas Conference on Information Systems*, 2008.

[3] A. Pentland, "How Social Networks Network Best." *Harvard Business Review, February*, 2009.

[4] B. N. Waber, D. O. Olguin, T. Kim, and A. Pentland, "Productivity Through Coffee Breaks: Changing Social Networks by Changing Break Structure." in *30th International Sunbelt Social Network Conference*, 2010.

[5] A. Madan, S. Moturu, D. Lazer, and A. Pentland, "Social sensing: obesity, unhealthy eating and exercise in face-to-face networks," in *ACM Wireless Health*, 2010, pp. 104–110.

[6] A. Madan, M. Cebrian, D. Lazer, and A. Pentland, "Social sensing for epidemiological behavior change," in *ACM UbiComp*, 2010, pp. 291–300.

[7] T. Kim, M. Chu, O. Brdiczka, and J. Begole, "Predicting shoppers' interest from social interactions using sociometric sensors," in *ACM CHI Extended Abstracts*, 2009, pp. 4513–4518.

[8] H. Lu, W. Pan, N. D. Lane, T. Choudhury, and A. T. Campbell, "Soundsense: scalable sound sensing for people-centric applications on mobile phones," in *ACM MobiSys*, 2009, pp. 165–178.

[9] Y. Wang, J. Lin, M. Annavaram, Q. Jacobson, J. Hong, B. Krishnamachari, and N. Sadeh, "A framework of energy efficient mobile sensing for automatic user state recognition."

[10] H. Lu, B. Brush, B. Priyantha, A. Karlson, and J. Liu, "Speakersense: Energy efficient unobtrusive speaker identification on mobile phones," in *Pervasive Computing*. Springer, 2011, pp. 188–205.

[11] P. Klasnja, S. Consolvo, T. Choudhury, R. Beckwith, and J. Hightower, "Exploring privacy concerns about personal sensing," *Pervasive Computing*, pp. 176–183, 2009.

[12] G. Iachello and K. Truong, "Prototyping and sampling experience to evaluate ubiquitous computing privacy in the real world," in *ACM CHI*, 2006, pp. 1009–1018.

[13] K. Plarre, A. Raij, S. Hossain, A. Ali, M. Nakajima, M. al'Absi, E. Ertin, T. Kamarck, S. Kumar, M. Scott, D. Siewiorek, A. Smailagic, and L. E. Wittmers, "Continuous Inference of Psychological Stress From Sensory Measurements Collected in the Natural Environment," in *ACM IPSN*, 2011.

[14] H. Lu, J. Yang, Z. Liu, N. Lane, T. Choudhury, and A. Campbell, "The Jigsaw continuous sensing engine for mobile phone applications," in *ACM SenSys*, 2010, pp. 71–84.

[15] M. Mun, S. Reddy, K. Shilton, N. Yau, J. Burke, D. Estrin, M. Hansen, E. Howard, R. West, and P. Boda, "Peir, the personal environmental impact report, as a platform for participatory sensing systems research," in *ACM MobiSys*, 2009, pp. 55–68.

[16] K. Plarre, A. Raij, S. Guha, and S. Kumar, "Automated detection of sensor detachments for physiological sensors in the wild," in *ACM Wireless Health*, 2010.

[17] D. Wyatt, T. Choudhury, J. Bilmes, and J. Kitts, "Inferring colocation and conversation networks from privacy-sensitive audio with implications for computational social science," *Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, no. 1, p. 7, 2011.

[18] Y. Lee, C. Min, C. Hwang, J. Lee, I. Hwang, Y. Ju, C. Yoo, M. Moon, U. Lee, and J. Song, "Sociophone: everyday face-to-face interaction monitoring platform using multi-phone sensor fusion," in *Proceeding of the 11th annual international conference on Mobile systems, applications, and services (MobiSys).* ACM, 2013, pp. 375–388.

[19] M. Rossi, O. Amft, S. Feese, C. Käslin, and G. Tröster, "Myconverse: recognising and visualising personal conversations using smartphones," in *Proceedings of the 2013 ACM conference on Pervasive and ubiquitous computing adjunct publication.* ACM, 2013, pp. 1275–1284.

[20] C. Luo and M. C. Chan, "Socialweaver: collaborative inference of human conversation networks using smartphones," in *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems (SenSys).* ACM, 2013, p. 20.

[21] C. Xu, S. Li, G. Liu, Y. Zhang, E. Miluzzo, Y.-F. Chen, J. Li, and B. Firner, "Crowd++: unsupervised speaker count with smartphones," in *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing (UbiComp).* ACM, 2013, pp. 43–52.

[22] P. Várady, T. Micsik, S. Benedek, and Z. Benyó, "A novel method for the detection of apnea and hypopnea events in respiration signals," *Biomedical Engineering, IEEE Transactions on*, vol. 49, no. 9, pp. 936–942, 2002.

[23] J. Han, H. Shin, D. Jeong, and K. Park, “Detection of apneic events from single channel nasal airflow using 2nd derivative method,” *Computer methods and programs in biomedicine*, vol. 91, no. 3, pp. 199–207, 2008.

[24] P. Rainville, A. Bechara, N. Naqvi, and A. Damasio, “Basic emotions are associated with distinct patterns of cardiorespiratory activity,” *International journal of psychophysiology*, vol. 61, no. 1, pp. 5–18, 2006.

[25] A. J. Wilson, C. I. Franks, and I. L. Freeston, “Algorithms for the detection of breaths from respiratory waveform recordings of infants,” *Medical and Biological Engineering and Computing*, vol. 20, no. 3, pp. 286–292, 1982.

[26] B. Todd and D. Andrews, “The Identification of Peaks in Physiological Signals,” *Computers and biomedical research*, vol. 32, no. 4, pp. 322–335, 1999.

[27] D. McFarland, “Respiratory markers of conversational interaction,” *Journal of Speech, Language, and Hearing Research*, vol. 44, no. 1, pp. 128–143, 2001.

[28] F. Wilhelm, E. Handke, and W. Roth, “Detection of speaking with a new respiratory inductive plethysmography system,” *Biomedical Sciences Instrumentation*, vol. 39, p. 136, 2003.

[29] M. Rahman, A. A. Ali, A. Raij, E. Ertin, M. al’Absi, and S. Kumar, “Online Detection of Speaking from Respiratory Measurements Collected in the Natural Environment,” in *ACM IPSN Demo Abstract*, 2011.

[30] E. Ertin, N. Stohs, S. Kumar, A. Raij, M. al’Absi, T. Kwon, S. Mitra, and S. Shah, “AutoSense: Unobtrusively Wearable Sensor Suite for Inferencing of Onset, Causality, and Consequences of Stress in the Field,” in *ACM SenSys*, 2011.

[31] http://www.fieldstream.org/.

[32] http://mathforum.org/library/drmath/view/52720.html.

[33] M. Hall, “Correlation-based Feature Selection for Machine Learning,” Ph.D. dissertation, The University of Waikato, 1999.

[34] I. Witten, *Weka: Practical machine learning tools and techniques with Java implementations.* Dept. of Computer Science, University of Waikato, 1999.

[35] I. Witten and E. Frank, *Data Mining: Practical machine learning tools and techniques.* Morgan Kaufmann Pub, 2005.

[36] G. Forney Jr, “The viterbi algorithm,” *Proceedings of the IEEE*, vol. 61, no. 3, pp. 268–278, 1973.

[37] T. Choudhury, “Sensing and modeling human networks,” Ph.D. dissertation, Massachusetts Institute of Technology, February 2004.