11-1-2016

Automatic Pain Assessment from Infants' Crying Sounds

Chih-Yun Pai
University of South Florida, tf930302@gmail.com

Automatic Pain Assessment from Infants' Crying Sounds

by

Chih-Yun Pai

A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science in Computer Science
Department of Computer Science and Engineering
College of Engineering
University of South Florida

Major Professor: Dmitry B. Goldgof, Ph.D.
Rangachar Kasturi, Ph.D.
Yu Sun, Ph.D.

Date of Approval:
October 21, 2016

Keywords: Whimpering, Vigorous Crying, K-Nearest Neighbors, Random Forests, Least Squares Support Vector Machines

ACKNOWLEDGMENTS

I am grateful to the help and cooperation in the data collection from Judy Zaritt, Marcia Kneusel, and the neonatal staff at Tampa General Hospital. I also appreciate to the parents who allowed us to record their babies in this study.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Crying is infants utilize to express their emotional state. It provides the parents and the nurses a criterion to understand infants' physiology state. Many researchers have analyzed infants' crying sounds to diagnose specific diseases or define the reasons for crying. This thesis presents an automatic crying level assessment system to classify infants' crying sounds that have been recorded under realistic conditions in the Neonatal Intensive Care Unit (NICU) as whimpering or vigorous crying. To analyze the crying signal, Welch's method and Linear Predictive Coding (LPC) are used to extract spectral features; the average and the standard deviation of the frequency signal and the maximum power spectral density are the other spectral features which are used in classification. For classification, three state-of-the-art classifiers, namely K-nearest Neighbors, Random Forests, and Least Squares Support Vector Machine are tested in this work, and the experimental result achieves the highest accuracy in classifying whimper and vigorous crying using the clean dataset is 90%, which is sampled with 10 seconds before scoring and 5 seconds after scoring and uses K-nearest neighbors as the classifier.

CHAPTER 1: INTRODUCTION

Adults can self-report their pain experience. But, how can an infant express how much pain s/he suffered? Crying is one way an infant nonverbally communicates with others; crying contains information about infant's status [1]. Nurses in the neonatal intensive care unit (NICU) traditionally use different pain scales to evaluate an infant's pain state. Along with other indicators, crying is used as the main indicator in several pain scales. However, crying evaluation is subjective. It can be easily affected by other factors or the observer's experience [2, 3]. Even during different observations, the same observer can give different results. Machine-based automatic crying evaluation is a good way to provide consistent assessment and to minimize biased appraisals. A machine-based automatic crying evaluator can also be used in monitoring infants in the NICU or house care. Furthermore, it is a good way to improve the quality of medical service in places which lack medical facilities.

1.1 Prior Works

Existing works in automatic analysis of infants' crying sounds focus on either determining the reason an infant cries [4-8] or pathological diagnosing [1, 9-14]. There is not aware of any research that assesses different levels of infants' cry. Xie et al. presents H-value, which is driven by the mode of crying representation in crying signal with a hidden Markov model based classifier to assess infants' level of distress [15]. Chang et al. extract 15 features from time domain and frequency domain in the incremental learning Support Vector Machines infant crying recognition system [4]. The system selects four features in estimating different causes of infant crying. The average accuracy in predicting why the infant cries is 85%, and the accuracy of estimating if the infant is in pain or no pain is 82%. Mima and Arakawa propose a rule-based system for classifying infants' crying reason between hunger, sleepiness, and discomfort [5]. They analyze the shape of power spectrum of crying signal and achieve 85% accuracy in classifying infants' crying reasons. Vempada et al. combine spectral and prosodic features and use them to train the crying pattern with Support Vector Machines (SVM) [6]. The recognition performance of using spectral features and prosodic features in detecting pain or no pain are 31% and 83%, and the performance in recognizing crying signal with pain by using two different type of features is 81%. Petroni et al. use Mel Frequency Cepstral Coefficients (MFCC) as the features in infants' crying classification with artificial neural networks [7]. They achieve the accuracy of 90% in predicting if an infant is in pain. Barajas-Montiel and Reyes-Garcia ensemble AdaBoost algorithm in Neural Network and SVM to classify the cries as pain or no pain and hunger or no huger [8]. They obtain 96% accuracy in classifying pain or no pain using ensemble Neural Network as classifier and MFCC as the feature.

Reyes-Galaviz and Reyes-Garcia compare the performance between extracting the features of using LPC (Linear Prediction Coding) and MFCC and classify with neural networks [1]. The result shows 76% precision in diagnosing normal, deaf, or asphyxia infant with LPC after 2,000 training epochs and 86% precision in diagnosis with MFCC after 1,414 training epochs. Zabidi et al. generate the model to distinguish infants with hypothyroidism from crying [9]. They extract MFCC as the feature and select it directly or with Fisher's Ratio (F-Ratio) analysis. Using Multilayer perceptron (MLP) neural network as classifier, they achieve 89% classification accuracy with performing F-Ratio in MFCC selection. Reyes-Galaviz et al. present an infant crying recognizer with feed forward input delay neural network to recognize normal cry and pathological cry from Cuban and Mexican infants [10]. Their experiment, which uses MFCC as the feature, obtain almost 100% accuracy in recognizing normal cry and pathological cry from Mexican infants. Saraswathy et al. try Probabilistic Neural Network (PNN) and General Regression Neural Network (GRNN) in classifying cries from normal infants, deaf infants, and infants with asphyxia [11]. 20 features are extracted from short-time Fourier transform. Both PNN and GNN achieve 99% classification results. Garcia and Reyes-Garcia present an infant crying recognition system using MFCC as the feature and feed forward neural network as classifier with several learning methods in training [12]. They aim to classify normal infants and deaf infants from their cries. Scaled Conjugate Gradient neural network results the better accuracy with 97% in their experiment. Lederman et al. use MFCC as the feature and classify with Continuous Density Hidden Markov Models [13]. They attempt to determine the cries are from the healthy infants or the infants who experienced RDS (Respiratory Distress Syndrome) and the infants with or without palatal plate. The diagnosis accuracy of the infants with RDF is 63%, and the mean correct classification of the infants with or without palate plate is 57% with subject independent tests (all cries are from the same age group). Santiago-Sanchez et al. present a type-2 fuzzy sets based pattern matching method for classifying infant crying [14]. This work uses cochleograms, intensity, LPC, and MFCC as the features and achieves 85%, 61%, 89%, and 79% precision respectively when classifying crying patterns between normal, asphyxia, and hyperbilirubinemia infant.

1.2 Crying Level Assessment System

This work presents an audio frequency based infant crying classification scheme. The goal is to provide the nurse an appraisal of the level of infant crying when assessing infants' pain state during a procedure. The infants crying dataset in this work was recorded during medical procedures in the NICU which includes significant noise, such as human speech, machine sound, knocking sound, and other infants' crying. I classify crying episodes as either whimpering or vigorous crying, and test using different sampling lengths of crying episode.

In the next chapter, I will introduce a speaker recognition system which is re-implemented in this work. The algorithm of the crying level assessment system is introduced in chapter 3. The infants crying dataset is described in chapter 4. Chapter 5 and chapter 6 are the experiment setup and the experimental results of both my method and the speaker recognition system. The summary and the discussion are listed in chapter 7.

CHAPTER 2: BACKGROUND

Yang and Jing propose a speaker recognition system using SVM-VQ [16]. Multiple features, which include pitch, LPCC, ΔLPCC, MFCC, and ΔMFCC, are extracted from the TIMIT speech database [17].

2.1 LPC, LPCC, and ΔLPCC

LPC is widely used in audio signal processing to represent the spectral features. It compresses the audio signal within the spectral envelope. LPC-derived cepstral coefficients (LPCC) is commonly used in short time period spectral analysis. An p order LPC system with n sampling points signal $s(n)$ is shown:

$$s(n) = \sum_{k=1}^{p} a_k s(n-k) + e(n). \tag{2.1}$$

$e(n)$ is prediction error, and a_k is the linear predictor coefficients. To express it in z field,

$$S(z) = \sum_{k=1}^{p} a_k z^{-k} S(z) + E(z), \tag{2.2}$$

and the system transfer function is calculated as

$$H(z) = \frac{S(z)}{E(z)} = \frac{1}{\sum_{k=1}^{p} a_k z^{-k}} = \frac{1}{A(z)}. \tag{2.3}$$

LPC can be solved by Levinson-Durbin algorithm.

$$\begin{cases} a_i^{(i)} = k_i \\ a_j^{(i)} = a_j^{(i-1)} - k_i a_{i-j}^{(i-1)}, \quad 1 \le j \le i \end{cases} \tag{2.4}$$

Once LPC coefficients are solved, LPCC can be derived

$$c_m = a_m + \sum_{k=1}^{m-1} \frac{k}{m} c_k a_{m-k}, \qquad 1 \le m \le p, \tag{2.5}$$

and the delta-cepstral coefficients (ΔLPCC) are calculated with

$$\Delta c_m(t) = \sum_{i=-1}^{1} i \times c_m(t+i), \qquad 1 \le m \le p. \tag{2.6}$$

2.2 MFCC and ΔMFCC

MFCC is a representation of power spectrum in short time period. It is based on the characteristics of the critical bandwidths. In extracting MFCC, the length of Hamming windows is 30ms with 10ms shifting. FFT is sampled with 1,024 points, and the number of Mel-filter bank is 24. An m order MFCC coefficients are obtained with logarithmic and Discrete Cosine Transform (DST) by

$$c_m = \sqrt{\frac{2}{N}} \sum_{k=1}^{N} \cos\left[m\frac{\pi}{N}\left(k - \frac{1}{2}\right)\right] \log_{10} X_k, \tag{2.7}$$

where N is the number of Mel-filters and X_k is the kth output of the filter. ΔMFCC can be computed by formula (2.6).

2.3 SVM-VQ

Lebrun, et al. first used Vector Quantization (VQ) with Support Vector Machines (SVM) to simplify the training set by mapping pixels to representative prototype [18]. Yu, et al. used SVM-VQ, which combines Vector Quantization and Support Vector Machines, to deal with high dimensioned imbalanced data by compressing the majority class [19]. It compresses the data using Vector Quantization and classifies with SVM.

2.3.1 Vector Quantization (VQ)

Vector Quantization is a classical signal quantization technique. It divide a set of vectors into groups with Nearest Neighbor condition. Each vector within a group can be represented by a prototype vector. Suppose M training vectors, $X = \{x_1, x_2, \dots, x_M\}$, and N code vectors, $CV = \{cv_1, cv_2, \dots, cv_N\}$, all training vectors can be grouped in N sub-regions, $S = \{S_1, S_2, \dots, S_N\}$.

Vectors x_m within region S_n can be represented by code vector cv_n, which also represents the center of region S_n. x_m should satisfy

$$S_n = \{x_m : \|x_m - cv_n\|^2 \leq \|x_m - cv_{n'}\|^2\}, \ \ \forall n' = 1,2,\ldots,N, \qquad (2.8)$$

and

$$cv_n^d = \frac{\sum_{x_m \in S_n} x_m^d}{\sum_{x_m \in S_n} 1}, \qquad \forall \mathrm{d} = 1,2,\ldots,D \qquad (2.9)$$

with D is the dimension of the vector. Hence, $cv_n = \{cv_n^1, cv_n^2, \ldots, cv_n^D\}$ and $x_m = \{x_m^1, x_m^2, \ldots, x_m^D\}$.

2.3.2 Support Vector Machines (SVM)

SVM transforms the low-dimensional nonlinear problem into a higher-dimensional linear problem. It can be treated as a linear learning machine in the higher-dimensional linear feature space. The quantized training vectors from VQ are used to build the training model in SVM with the vector of class labels, $Y = \{y_1, y_2, \ldots, y_M\} \in \{-1,1\}$. The optimization problem of SVM is written as

$$\min_{w,\xi,b} J_1(w,\xi) = \frac{1}{2} w^T w + C \sum_{i=1}^{M} \xi_i, \qquad (2.10)$$

which subject to $y_i(w^T \varphi(x_i) + b) \geq 1 - \xi_i$ and $\xi_i \geq 0$ for $i = 1,\ldots,M$, where $\varphi(x_i)$ is a mapping function which converts vector x_i to a high-dimensional feature space, w is an unknown vector with the same dimension as $\varphi(x_i)$, $C > 0$ defines the trade-off between a large margin and classification error in the cost function, and ξ_i indicates the distance between x_i and the decision boundary. The decision function of SVM is formed as

$$f(x) = sign\left(\sum_{i=1}^{\#SV} y_i \alpha_i K(x, x_i) + b\right), \qquad (2.11)$$

where $\#SV$ represents the number of support vectors, α_i specifies the coefficients of the hyperplane, and $K(x, x_i)$ is the kernel function.

2.3.3 Least Squares Support Vector Machines (LS-SVM)

Instead of solving a quadratic problem and using unequal constraints in SVM, LS-SVM uses a least squares loss function and equality constraint to reduce the complexity [20, 21]. Different from the optimization problem of SVM (formula 2.10), LS-SVM optimizes the problem:

$$\min_{w,e,b} J_2(w,e) = \frac{1}{2}w^T w + \gamma \frac{1}{2}\sum_{i=1}^{M} {e_i}^2, \tag{2.12}$$

subject to $y_i(w^T\varphi(x_i) + b) \geq 1 - e_i$ for $i = 1, \dots, M$, where e_i is the error variable to tolerate misclassification and γ is the positive regularization constant. LS-SVM classifier is given by

$$f(x) = sign(\sum_{i=1}^{M} y_i \alpha_i K(x, x_i) + b), \tag{2.13}$$

which is similar to SVM (formula 2.11).

2.4 Yang's Speaker Recognition System

Instead of using SVM, Yang's speaker recognition system compresses the features with trained codebook using vector quantization and classifies the compressed feature using LS-SVM. In extracting LPCC, Hamming window of 32ms is used that shifts every 16ms. Yang and Jing compare the performance when using different features. Their experiment shows that using LPCC (92% recognition rate) or MFCC (90% recognition rate) alone as the feature has higher performance. Combining pitch with LPCC or MFCC, the recognition rate has only 1% improvement in recognizing the speaker [16].

In this work, I will implement Yang's speaker recognition system. The codebook size is 32, and the codebook is trained with the infants crying dataset which is used in this work. The performance of classifying the infants crying dataset with Yang's speaker recognition system is compared with the proposed crying level assessment system.

CHAPTER 3: PROPOSED ALGORITHM

In this work, I extract LPC, mean, standard deviation (STD), and the maximum power spectral density (P_{max}) as the features with frequency analysis and power spectrum analysis. Figure 3.1 shows the system schema of the crying level assessment system. Experiments on the input signal are performed with two different sizes of windows.

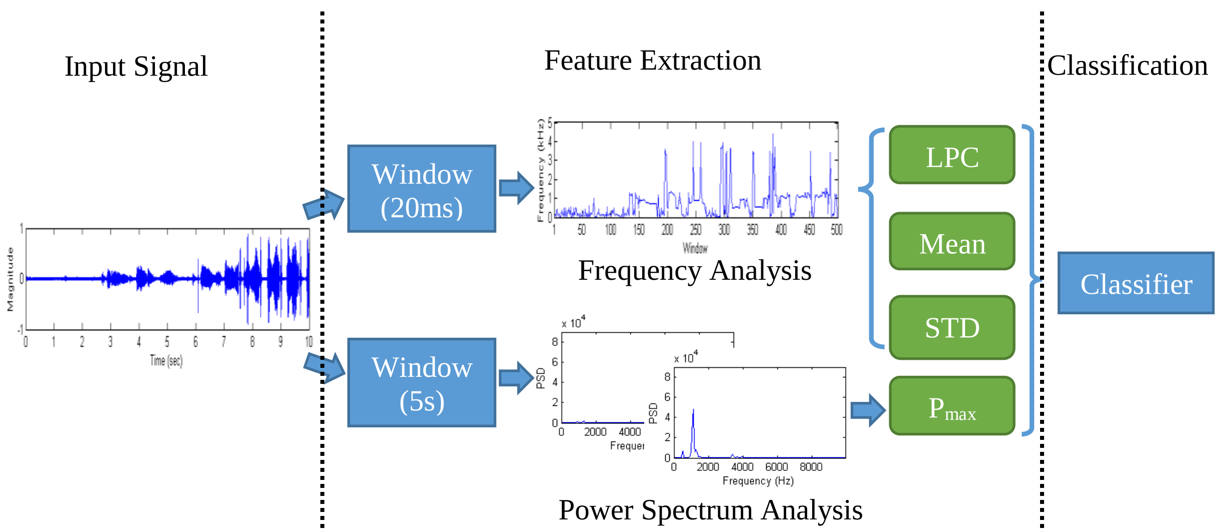

Figure 3.1 System schema

3.1 Windowing Signal

Audio signals are stable in short intervals [4]. Hence, I divide the segmented audio signal (episode) into several consecutive 20-millisecond windows. Frequency analysis is performed on each 20-millisecond window individually. In power spectrum analysis, I use 5-second windows to analyze the power spectral density in a longer time period. For each episode, I have performed two different windowing schemes. The first scheme is without overlapping between consecutive windows, and the second one has 50%-overlapping on the two neighboring windows.

3.2 Feature Extraction

To extract the feature vectors, I provide frequency analysis with short intervals and power spectrum analysis with a longer time period to extract the features. The frequency with the maximum energy is analyzed in frequency analysis.

3.2.1 Frequency Analysis

An input signal is divided into several consecutive 20-millisecond windows. Each window will be analyzed with Welch's Method [22] and have a frequency with the highest power spectral density. With these frequencies from the windows, I extract the spectral features and the linear predictive coefficients.

Welch's method divides the time signal into blocks and forms the periodogram. The mth window from the signal x can be denoted as

$$x_m(n) \triangleq w(n)x(n+mR), n = 0,1, \dots, M-1, m = 0,1, \dots, K-1, \qquad (3.1)$$

where R is the window shifting size, $w(n)$ is the window function which contains M nonzero samples, and K is the number of available frames. The periodogram can be shown as

$$P_{x_m,M}(w_k) = \frac{1}{M}\left|FFT_{N,k}(x_m)\right|^2 \triangleq \frac{1}{M}\left|\sum_{n=0}^{N-1} x_n(n)e^{-j^{2\pi nk/N}}\right|^2, \qquad (3.2)$$

and the estimated power spectral density is given by

$$\hat{S}_x^W(w_k) \triangleq \frac{1}{K}\sum_{m=0}^{K-1} P_{x_m,M}(w_k). \qquad (3.3)$$

Specifically, each window is analyzed with Welch's Method to extract the frequency with the highest power. Figure 3.2 illustrates the procedure of frequency analysis, and the windowing scheme without overlapping. Each episode will generate a frequency sequence which contains all the frequencies extracted from the windows. Figure 3.2 (a) shows two 20-millisecond windows out of a total of 500 windows over a 10-second audio signal (episode). The power spectral density (PSD) of frequencies in a window is analyzed with Welch's method, and is shown in Figure 3.2 (b). I extract the frequency with the maximum power density which is the peak in the window. Figure 3.2 (c) is the extracted frequency sequence which has 500 frequencies of an episode. I also compute the mean and the standard deviation of the frequencies in each episode as spectral features. Two frequency sequences are generated in frequency extraction, one with non-overlapping windows and one with 50%-overlapping windows. The spectral features, which are the mean and standard deviation, were also extracted from the non-overlapping and 50%-overlapping windows of episodes.

LPC coefficients can be derived by formulas 2.1 to 2.4. Instead of extracting a set of coefficients in each window, which is normally used in audio signal processing, I extract LPC coefficients from each frequency sequence I described above.

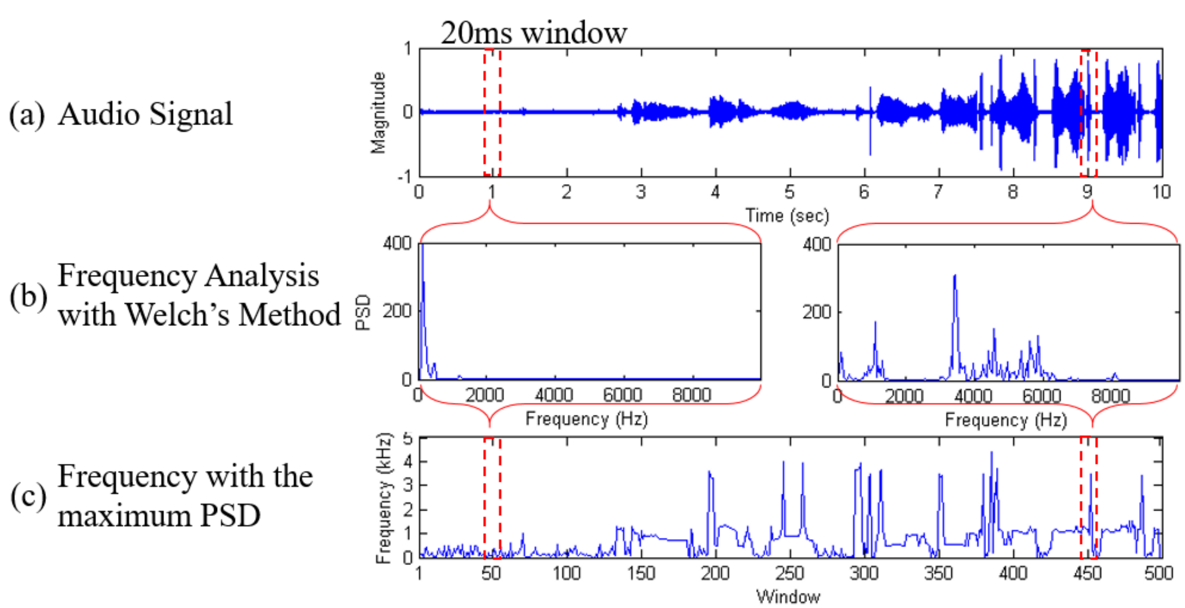

Figure 3.2 Schematic diagram of frequency analysis. (a) Audio signal (10-second episode) with two 20-millisecond-windows (b) The power spectral density of each frequency in two windows with frequency analysis using Welch's method (c) Frequencies with the maximum PSD of all windows in an episode (500 windows)

3.2.2 Power Spectrum Analysis

Instead of analyzing the signal features in short interval, power spectrum analysis attempt to analyze the PSD of frequencies in a longer period. Since the noises with high PSD usually appear in the short time period, and infants' crying sounds continue longer than the noises, the segments with crying sounds can be distinguished with the higher PSD in a larger window. In each 5-second window, I utilize Welch's method to analyze the power spectral density and get the maximum PSD since the crying signal should be louder than others.

3.2.3 Feature Vectors

Table 3.1 lists the feature vectors with non-overlapping window schema and 50%-overlapping window schema. The different numbers of LPC coefficients are tested in the experiment to determine the higher classification accuracy. The testing range of the coefficients is from 10 to 30. Mean and standard deviation of frequency sequence are two single-features, and the number of maximum power spectral densities are varied by the lengths of episode and the windowing scheme.

Table 3.1 Feature vectors

V1	Non-overlapping windows	LPC coefficient sequence Mean of frequency sequence Standard deviation of frequency sequence Sequence of maximum power spectral density
V2	50%-overlapping windows	LPC coefficient sequence Mean of frequency sequence Standard deviation of frequency sequence Sequence of maximum power spectral density

CHAPTER 4: INFANTS CRYING DATASET

The data is recorded and collected in Tampa General Hospital using GoPro Hero 3 plus, and the audio is extracted using VLC media player as the infants crying dataset. The audio sampling rate is 48 kHz. Subjects' average age is 36 gestational weeks. This dataset has a total of 27 subjects which are recorded during 32 acute painful procedures, and the total number of episodes with the ground truth from nurses is 128.

4.1 Data Collection

Each sample is recorded under an acute painful procedure, and the pain assessments are given in seven time periods (episodes):

- 5 minutes before procedure to be the baseline.
- Start the procedure.
- 1 minute after completing the procedure.
- 2 minute after completing the procedure.
- 3 minute after completing the procedure.
- 4 minute after completing the procedure.
- 5 minute after completing the procedure.

Neonatal Infant Pain Scale (NIPS) which has six pain indicators [23] is used in assessing infants' pain state in this project and is given by the expert nurses as the ground truth of each episode. In NIPS, crying indicator can be scored as 0 (no cry), 1 (whimper), and 2 (vigorous crying). Since this work only focuses on classifying whimpering and vigorous crying episodes, 14 whimpering episodes and 20 vigorous crying episodes are used in this work; other episodes do not have crying sounds and thus are excluded from further analysis. Each episode is extracted with eight sub-samples which are the different time interval combinations. They are sampled with 0, 5, 10, or 15 seconds before the nurse giving the score and 5 or 10 seconds after scoring. Summary of these intervals is given in Table 4.1, and SS1 to SS8 are eight sub-samples of an episode. Normally, the nurse observes the infant for 15 seconds before giving the score, and the infant state may change within 5 seconds, which is SS4.

Table 4.1 Time interval combinations of sub-sample

	Before				After	
	0 sec	5 sec	10 sec	15 sec	5 sec	10 sec
SS1	V				V	
SS2		V			V	
SS3			V		V	
SS4				V	V	
SS5	V					V
SS6		V				V
SS7			V			V
SS8				V		V

4.2 Clean Data and Additional Ground Truth

Since some episodes of the original infants crying dataset are too noisy and will affect the classification, I eliminated 13 noisy episodes to get a clean dataset. Also, to enlarge the dataset, I use the original dataset and add additional ground truths to new episodes which are sampled every 20 seconds between two scored episodes in the original dataset. In Figure 4.1, episode 1 and episode 2 are sampled from the original dataset and episode 1-1 and episode 1-2 are the additional episodes between episode 1 and episode 2. In this larger dataset, I only keep the episodes with the last 5 seconds which are the time giving score with no noise. The first 15 seconds (the observation time) can be either noisy or quiet. The scores of the additional episodes which do not have score in the original dataset are not given by the trained nurses. As seen in Figure 4.1, only episode 1 and episode 2 are scored by the nurses. Following the same scoring procedure (observe the infant for 15 seconds and give the score), I label episode 1-1 and episode 1-2 as either whimpering or vigorous crying. The number of episodes of the original dataset, the clean dataset, and the additional dataset are listed in Table 4.2.

Table 4.2 Number of episodes of each class in each dataset

	Original	Clean	Additional
Whimper	14	7	71
Vigorous crying	20	14	94
Total	34	21	165

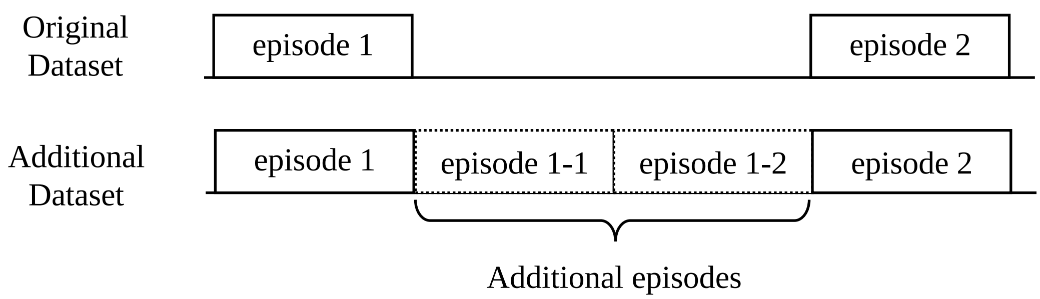

Figure 4.1 Example of having additional episodes

CHAPTER 5: EXPERIMENT SETUP

This chapter will introduce the classifiers and classification strategy. The experiment of the crying level assessment system and Yang's speaker recognition system is implemented with Matlab 2013b.

5.1 Classifiers

In this work, I use three classifiers, K-nearest Neighbors (KNN), Random Forests, and Least Squares Support Vector Machines (LS-SVM). 10 to 30 LPC coefficients and different K's (1 to 9) are tested in training step to determine the best number of coefficients.

5.1.1 K-nearest Neighbors

K-nearest-neighbor is a simple nonparametric classification method [24]. It classifies a sample to the majority class which is observed from the kth nearest neighbors in the feature space. The time complexity of training K-nearest neighbors model is $O(nd + kn)$, where n, d, and k are the number of instances, data dimension, and the number of neighbors, respectively. In this work, I train k from one to nine.

5.1.2 Random Forests

Random Forests is an extension of machine learning classifier which include the bagging to improve the performance of Decision Tree. It combines tree predictors, and trees are depended on a random vector which is independently sampled. The distribution of all trees are the same. Random Forests splits nodes using the best among of a predictor subset that are randomly chosen from the node itself, instead of splitting nodes based on the variables [25, 26, 27]. The time complexity of the worst case of learning with Random Forests is $O(M(dn\log n))$, where M is the number of growing trees, n is the number of instances, and d is the data dimension. In this work, the number of growing trees is 100.

5.1.3 Least Squares Support Vector Machines

Support vector machine (SVM) is a powerful machine learning tool and has been wildly used in pattern recognition. LS-SVM is a least squares version of SVM. LS-SVM considers the equality constraints using a quadratic error function. Instead of solving quadratic programming, LS-SVM solves the system of linear formulas. The time complexity of LS-SVM is $O(\min(n,d)^3)$, where n and d are number of instances and data dimension [20, 21, 28, 29]. In this work, both my method and Yang's method use linear kernel.

5.2 Model Evaluation

In this experiment, 10-fold cross validation is used as the strategy. In each training folds, I do parameters selection to decide the number of extracted coefficients in LPC for all classifiers and K for K-nearest Neighbors.

5.2.1 10-fold Cross Validation

10-fold cross validation splits instances into 10 folds (1 for testing set and 9 for training set). I perform ten 10-fold cross validation on all subjects (1st level 10-fold CV) and average the results. Since I want to determine the better number of LPC coefficients and K for K-nearest Neighbors, I do another 10-fold cross validation in training set (2nd level 10-fold CV) for parameter selection. Parameter selection is done by each pair of testing set and training set of 1st level 10-fold CV. Then the parameters will be fed back to the 1st level to do classification. Figure 5.1 is the pseudo-code of the 10-fold cross validation with parameter selection procedure.

1st level 10-fold CV

1. Testing set (1 fold): pick one fold
2. Training set (9 folds): remaining folds

   2nd level 10-fold CV (parameter selection)

   2.1. Spilt training set into 10 folds

   2.2. Try different number of LPC coefficients (Ns) in feature extraction

   2.3. Do classification and record the performance (If using K-nearest Neighbors, try different Ks.)

   2.4. Find out the parameters (N and K) with the highest performance

   End

3. Extract feature vector with trained N on both testing set and training set
4. Do classification

End

Figure 5.1 10-fold cross validation with parameter selection procedure

5.2.2 10 Folds Selecting Procedure

In the infants crying dataset, each sample (procedure) has different number of episodes with different classes, and some subjects are recorded in multiple samples. I use a procedure to split the instances (samples) into 10 folds by subject. In order to limit the overfitting problem, instances which belong to the same subject will be assigned to only one fold. The episodes of each class are evenly assigned to each fold with this procedure.

Figure 5.2 shows the flow chart of the procedure. All subjects are in the same group before starting the procedure. I first count the episodes of each class in all subjects. Select an empty fold, then I assign all instances which belong to the subject with the most episodes in any class to the selected fold. Compute the number of episodes of each class in the selected fold. To choose the next subject, I generate a candidate list which includes all the unselected subjects. Next, I filter out the subject from the candidate list with the following rules in order: (Fold limitation is 10% of the instances. Class limitations are 10% of the episodes which are labeled as each class. Each class has its own class limitation.)

(a) The selected fold should not reach the class limitations. If the numbers of episodes of every class in the selected fold reach class limitations, but the fold does not reach the fold limitation, mark the fold 'not-full.' Then I stop assigning subjects to this fold and find the next empty fold.

(b) If it is the last space in the selected fold (one more instance to reach fold limitation), and all the episodes in this fold are labeled as the same class, filter out all subjects which do not have the missing class or have multiple instances. Then skip rule (c).

(c) Check the class limitation on each class in the selected fold. Filter out the subjects which will make the selected fold exceed the class limitations.

(d) The candidate list can not be empty. If it is empty, mark the selected fold as 'need-class,' stop assigning subjects to this fold, and find the next empty fold.

If the selected fold passes rule (a) and (d), randomly select a subject in the candidate list, and assign it to the fold. Make a new candidate list and do the filtering again until the number of subjects in the selected fold reach the fold limitation, or the fold violate rule (a) or (d). Pick another empty fold and continue the subject assigning procedure until there is no empty fold. If there are un-assigned subjects but no empty folds, I assign them based on the insufficient class of the folds. Check the 'need-class' fold, and randomly assign the subjects which have the episodes with the class that the fold needs, if it does not reach the fold limitation. The ‘not-full’ folds which reach all class limitations but does not reach the fold limitation will be filled with the remaining subjects in the final step.

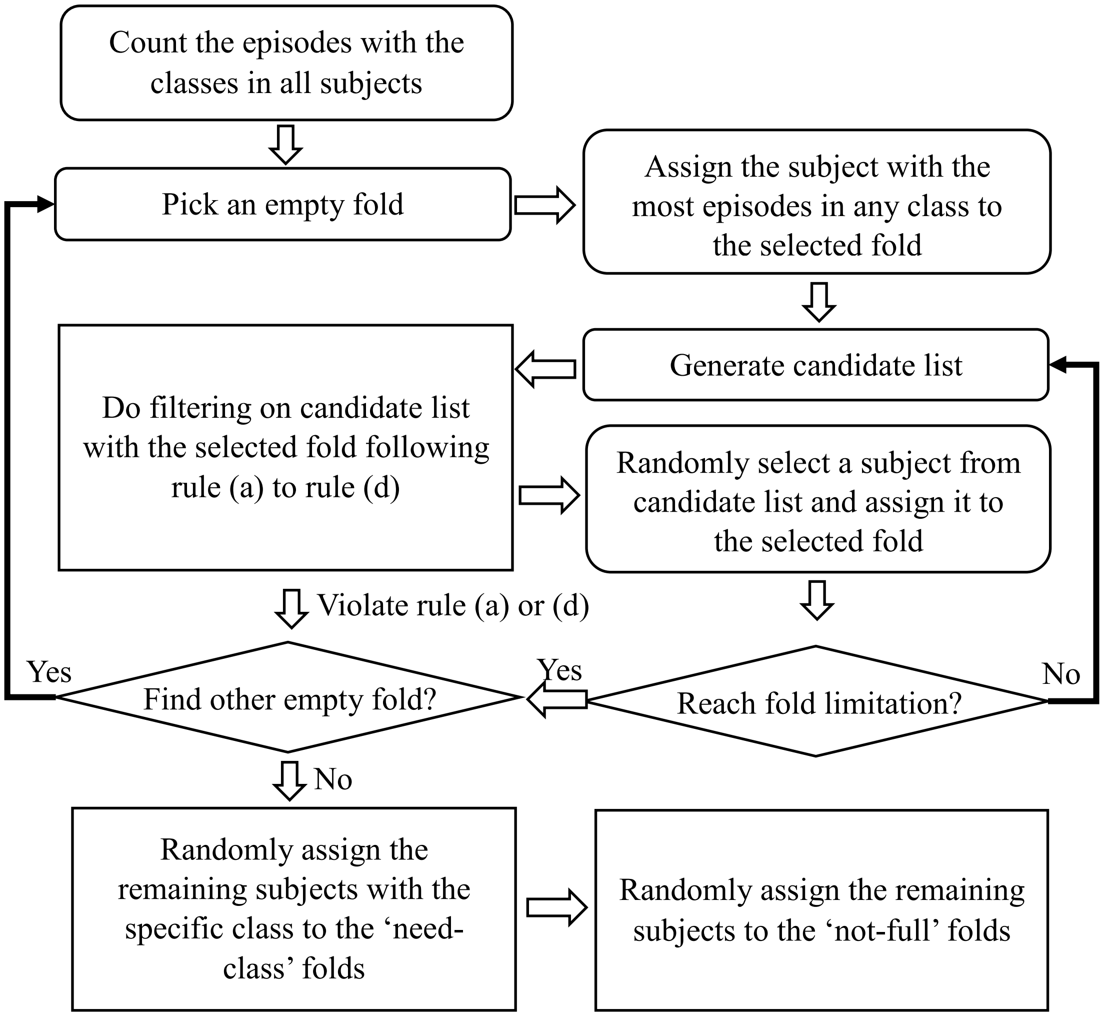

Figure 5.2 Flow chart of 10 folds selecting procedure. (Each subject has multiple episodes, and each episode is labeled as either whimpering or vigorous crying)

CHAPTER 6: EXPERIMENTAL RESULTS

This section will report the experimental results with three classifiers in my crying level assessment system and the results of classifying the infants crying dataset with Yang's speaker recognition system [16].

6.1 Results of Original Dataset

Table 6.1 lists the accuracies of classifying with K-nearest Neighbors (KNN), Random Forests, and Support vector machine (SVM). V1 and V2 are the feature vectors with non-overlapping window schema and 50%-overlapping window schema. SS1 to SS8 are eight different sampling length of an episode. The number of parameters of each feature vector are listed in Table 6.2, and Table 6.3 lists K's for K-nearest Neighbors. The highest accuracy of classifying infant crying level under realistic conditions is 76.47%, which adapts with K-nearest Neighbors with SS7, which is the sampling length combining 10 seconds before giving score and 10 seconds after giving score, and V1 (feature vector with non-overlapping windowing scheme). The performance with the highest classification accuracy which uses KNN with V1 and SS7 is shown in Figure 6.1 using receiver operating characteristic (ROC curve). Table 6.4 shows the performance of the highest accuracy in each classifier. Also, comparing recall, my method has higher performance to predicting vigorous crying than whimpering.

Table 6.1 Classification accuracy of the original dataset

	KNN		Random Forests		LS-SVM	
	V1	V2	V1	V2	V1	V2
SS1	66.47%	62.35%	61.76%	53.53%	65.29%	53.24%
SS2	66.18%	73.82%	74.41%	72.94%	42.65%	59.12%
SS3	60.59%	65.88%	72.65%	71.76%	62.65%	48.24%
SS4	72.35%	70.59%	65.29%	66.47%	63.53%	47.94%
SS5	57.65%	62.35%	67.06%	70.00%	68.24%	58.82%
SS6	73.82%	72.35%	72.35%	72.06%	**69.71%**	64.12%
SS7	**76.47%**	74.41%	**75.88%**	71.18%	67.65%	43.82%
SS8	61.47%	61.18%	68.24%	65.59%	66.47%	47.06%

Table 6.2 Number of parameters of three classifiers with the original dataset

	KNN		Random Forests		LS-SVM	
	V1	V2	V1	V2	V1	V2
SS1	13	13	17	17	21	13
SS2	14	15	31	21	31	34
SS3	15	17	18	22	33	28
SS4	16	19	16	26	21	36
SS5	14	15	15	17	17	29
SS6	15	17	26	26	**34**	37
SS7	**16**	19	**35**	27	20	31
SS8	17	21	20	21	22	30

Table 6.3 K for K-nearest neighbors with the original dataset

	KNN	
	V1	V2
SS1	3	6
SS2	7	3
SS3	3	4
SS4	8	3
SS5	4	5
SS6	5	5
SS7	**7**	6
SS8	7	8

Table 6.4 Performance of the highest accuracy in each classifier with the original dataset

	KNN		Random Forests		LS-SVM	
	Whimper	Vigorous Crying	Whimper	Vigorous Crying	Whimper	Vigorous Crying
Recall	64.29%	85.00%	60.71%	86.50%	65.00%	73.00%
Precision	75.00%	77.27%	75.89%	75.88%	62.76%	74.87%

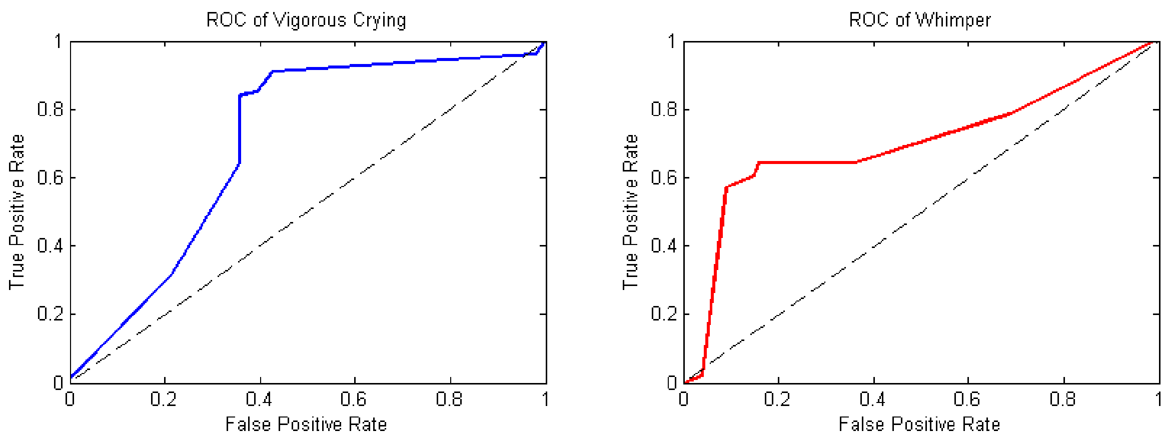

Figure 6.1 ROC curves with the original dataset using KNN with V1 and SS7

6.2 Results of Clean Dataset

Table 6.5 is the classification result with the clean dataset, which is smaller than the original dataset. The number of parameters and K for K-nearest Neighbors are listed in Table 6.6 and Table 6.7. Adapting with K-nearest Neighbors using SS3 and V1 achieves the highest accuracy (90%) in classifying infant crying with the clean dataset. The ROC curves of KNN with V1 and SS3 is shown in Figure 6.2, and the performance is shown in Table 6.8.

Comparing the best result in the original dataset (76.47%) with the clean dataset (90%), using the clean dataset has significant improvement in the classification accuracy. It points out that ambient noises have significant effects on classifying infant crying as whimpering or vigorous crying. Also, using the clean dataset increases the performance (recall and precision) in all classifiers except LS-SVM. Recall (sensitivity) of vigorous crying is increased in using the clean dataset.

Table 6.5 Classification accuracy of the clean dataset

	KNN		Random Forests		LS-SVM	
	V1	V2	V1	V2	V1	V2
SS1	77.14%	66.19%	69.05%	61.90%	55.24%	50.48%
SS2	80.48%	74.76%	79.52%	70.48%	56.19%	38.10%
SS3	**90.00%**	72.86%	75.71%	72.38%	60.48%	58.57%
SS4	76.19%	85.71%	80.00%	78.57%	69.52%	62.86%
SS5	70.00%	66.19%	70.95%	73.81%	55.71%	65.24%
SS6	51.90%	74.76%	79.52%	78.10%	54.29%	53.81%
SS7	77.62%	79.52%	**84.29%**	77.62%	60.95%	**71.90%**
SS8	71.43%	86.19%	80.00%	80.48%	68.10%	66.67%

Table 6.6 Number of parameters of three classifiers with the clean dataset

	KNN		Random Forests		LS-SVM	
	V1	V2	V1	V2	V1	V2
SS1	13	13	13	16	17	25
SS2	14	15	16	17	15	27
SS3	**15**	17	18	23	30	29
SS4	16	19	16	37	31	32
SS5	14	15	15	31	14	27
SS6	15	17	19	20	34	36
SS7	16	19	**17**	28	36	**31**
SS8	17	21	17	35	29	38

Table 6.7 K for K-nearest neighbors with the clean dataset

	KNN	
	V1	V2
SS1	7	4
SS2	5	3
SS3	**4**	5
SS4	4	4
SS5	4	7
SS6	5	3
SS7	3	4
SS8	3	6

Table 6.8 Performance of the highest accuracy in each classifier with the clean dataset

	KNN		Random Forests		LS-SVM	
	Whimper	Vigorous Crying	Whimper	Vigorous Crying	Whimper	Vigorous Crying
Recall	85.71%	92.14%	68.57%	92.14%	42.86%	86.43%
Precision	84.51%	92.81%	81.36%	85.43%	61.22%	75.16%

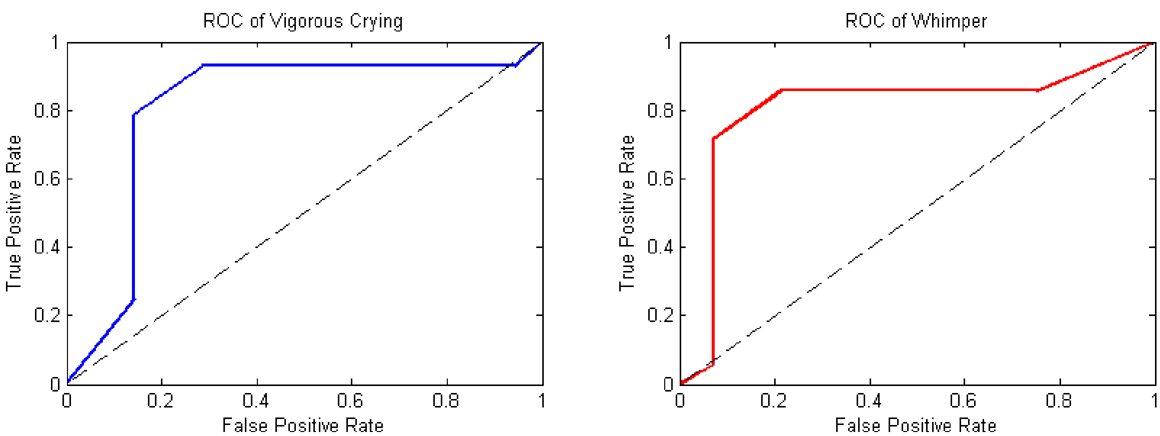

Figure 6.2 ROC curves with the clean dataset using KNN with V1 and SS3

6.3 Results of Additional Dataset

Tables 6.9, 6.10, and 6.11 show the classification results with the additional dataset (larger dataset), number of parameters, and K for K-nearest Neighbors, respectively. The highest accuracy in classifying infant crying is 78.85% using LS-SVM. The ROC curves of LS-SVM with V1 and SS1 are shown in Figure 6.3, and recall (sensitivity) and precision of classifying whimper and vigorous crying are listed in Table 6.12.

In the additional dataset, short sampling length episodes have higher performance than long episodes. It is caused by longer episodes that containing more noises (non-crying sound) have the significant impact in classifying crying. Compared with the result of the original dataset, with my method, adapting with K-nearest Neighbors has higher accuracy in classifying whimpering and vigorous crying with small dataset, and LS-SVM works better with large dataset. For the performance, the additional dataset raises the recall (sensitivity) of whimper, especially classifying with LS-SVM.

Table 6.9 Classification accuracy of the additional dataset

	KNN		Random Forests		LS-SVM	
	V1	V2	V1	V2	V1	V2
SS1	70.30%	70.97%	72.61%	**74.30%**	**78.85%**	75.33%
SS2	**70.79%**	66.00%	67.52%	68.24%	71.82%	70.30%
SS3	70.18%	66.85%	70.85%	67.39%	70.85%	71.88%
SS4	61.94%	55.76%	70.30%	68.85%	70.73%	73.39%
SS5	64.85%	60.00%	66.00%	67.45%	72.55%	71.94%
SS6	60.61%	64.00%	67.39%	66.30%	70.91%	70.18%
SS7	66.55%	68.30%	69.45%	71.03%	72.36%	74.06%
SS8	59.03%	61.45%	67.52%	68.79%	70.61%	72.36%

Table 6.10 Number of parameters of three classifiers with the additional dataset

	KNN		Random Forests		LS-SVM	
	V1	V2	V1	V2	15	17
SS1	13	13	17	**17**	**18**	25
SS2	**14**	15	33	27	17	24
SS3	15	17	25	26	16	24
SS4	16	19	18	19	14	15
SS5	14	15	19	25	25	17
SS6	15	17	25	33	16	27
SS7	16	19	28	37	18	23
SS8	17	21	32	25	15	17

Table 6.11 K for K-nearest neighbors with the additional dataset

	KNN	
	V1	V2
SS1	6	8
SS2	**8**	3
SS3	5	4
SS4	5	5
SS5	9	7
SS6	5	9
SS7	9	4
SS8	9	5

Table 6.12 Performance of the highest accuracy in each classifier with the additional dataset

	KNN		Random Forests		LS-SVM	
	Whimper	Vigorous Crying	Whimper	Vigorous Crying	Whimper	Vigorous Crying
Recall	63.94%	75.96%	66.20%	80.43%	79.72%	78.19%
Precision	66.76%	73.61%	71.87%	75.90%	73.41%	83.62%

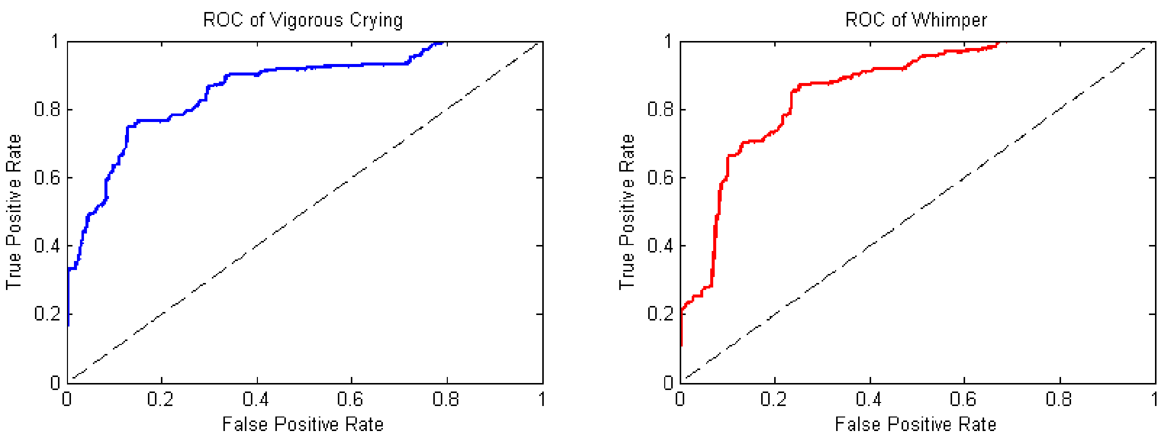

Figure 6.3 ROC curves with the additional dataset using LS-SVM with V1 and SS1

6.4 Comparing Results of Yang’s Speaker Recognition System

Table 6.13 shows the accuracy in classifying whimpering and vigorous crying using Yang’s speaker recognition system with the infants crying dataset. The ROC curves which correspond to the highest classification accuracy are shown in Figure 6.4. Recall (sensitivity) and precision of classification with the highest accuracy of each dataset are presented in Table 6.14. The accuracy of classifying whimpering and vigorous crying with the additional dataset is much lower than the other two datasets. The reason for this situation might be the feature compression. Since some whimpering episodes are similar to vigorous crying episodes, some information which could help the classifier to distinguish whimpering and vigorous crying is lost in the feature compression. Yang’s method does not work well in the additional dataset.

Yang’s method has high recall (sensitivity) for vigorous crying, even higher than my method. The highest accuracy of classifying whimpering and vigorous crying using the original dataset with MFCC (80.29%) and the clean dataset with LPCC (90.48%) are slightly higher than my method (76.47% while using the original dataset and 90.00% while using the clean dataset). However, my method has significant improvement in using the additional dataset which is larger than the other two. Also, my crying level assessment system adapting with K-nearest Neighbors (85.71%) has higher recall for whimpering than Yang's speaker recognition system (71.43%) in classifying with the clean dataset.

Table 6.13 Classification accuracy with Yang's speaker recognition system

	Original Dataset		Clean Dataset		Additional Dataset	
	LPCC	MFCC	LPCC	MFCC	LPCC	MFCC
SS1	75.88%	75.53%	88.10%	76.19%	50.85%	65.45%
SS2	72.06%	70.59%	**90.48%**	81.43%	46.97%	66.61%
SS3	73.53%	47.65%	85.71%	80.95%	62.12%	66.30%
SS4	64.71%	73.53%	80.95%	76.19%	61.21%	58.61%
SS5	77.06%	**80.29%**	81.43%	85.71%	62.42%	58.85%
SS6	73.24%	68.24%	80.95%	81.43%	63.94%	**69.76%**
SS7	76.47%	74.12%	85.71%	77.14%	45.88%	68.48%
SS8	76.47%	73.53%	85.71%	80.95%	48.30%	60.42%

Table 6.14 Performance of the highest accuracy of Yang's speaker recognition system

	Original Dataset		Clean Dataset		Additional Dataset	
	Whimper	Vigorous Crying	Whimper	Vigorous Crying	Whimper	Vigorous Crying
Recall	66.43%	90.00%	71.43%	100.00%	56.34%	79.89%
Precision	82.30%	79.30%	100.00%	87.50%	67.91%	70.78%

(a)

(b)

(c)

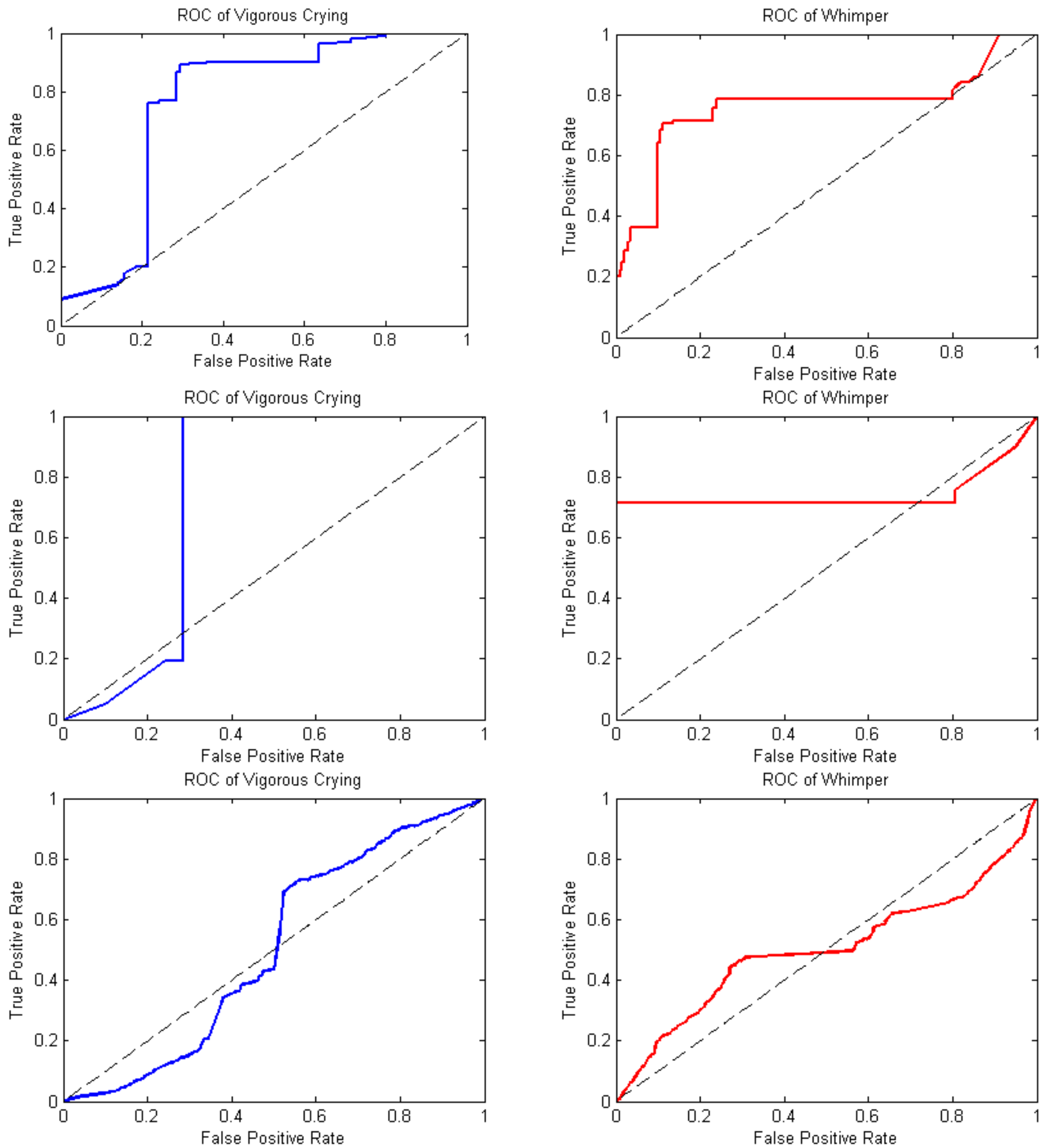

Figure 6.4 ROC curves of Yang's speaker recognition system. (a) the original dataset, (b) the clean dataset, and (c) the additional dataset

CHAPTER 7: SUMMARY AND DISCUSSION

Assessing the level of infants' crying is subjective and can be inconsistent. Significant ambient noises which are recorded during a medical procedure affects classifying performance. In this work, I present an automatic crying level assessment system which adapts with K-nearest Neighbors, Random Forests, and Least Squares Support Vector Machines to classify whimpering and vigorous crying signals under the realistic conditions in NICU. Three different sizes of dataset, which are the original dataset, the clean dataset, and the additional dataset, are used in this work. The highest accuracies of classifying infant crying signal in whimpering and vigorous crying are 76.47% which classifies with K-nearest Neighbors using the original dataset, 90.00% with K-nearest Neighbors using the clean dataset, and 78.85% with Least Squares Support Vector Machines using the additional dataset. Recall (sensitivity) for vigorous crying is higher than whimpering in my method.

In comparison, Yang's speaker recognition system achieves 80.29% accuracy in classifying whimpering and vigorous crying with the original dataset, 90.48% with the clean dataset, and 69.76% with the additional dataset. Since Yang's method is more sensitive in classifying vigorous crying and the original dataset and the clean dataset have higher ratio of vigorous crying and whimpering, the classification accuracies are slightly lower when using my method to classify these two dataset. But, my crying level assessment system has significant improvement in the additional dataset, which is larger than the other two and with more equally ratio in two classes. Also, my method has higher performance in classifying whimper than Yang's method.

7.1 Future Research

In this thesis, not crying is not considered and is filtered out manually. In the future, not crying class is going to add in the crying level assessment model. Since determining infants' crying reasons (including in pain) has been discussed [4-8], the model of pain detection with crying can be added as the preprocessing to extract crying episodes. Also, this work will combine with other project which using other indicators from NIPS, and assess infants' pain with multiple indicators. More subjects are enrolled and collected. Data of approximately 300 infants during acute (few minutes) and chronic (more than three hours) painful procedures is planned to collect to enlarge the infants crying dataset.

Next, I am going to implement this model in long term monitoring with the infants under chronic painful procedure. In order to use as a monitoring system, the other conditions that lead to infants' crying episodes should be considered. Infants cry not only for pain but also for other reasons, such as hunger and wet diaper. Adding these crying models in this system can provide better association for nurses in monitoring infants.

REFERENCES

[1] O. F. Reyes-Galaviz and C. A. Reyes-Garcia, "A system for the processing of infant cry to recognize pathologies in recently born babies with neural networks," in 9th Conf. on Speech and Computer (SPECOM'2004), St. Petersburg, Russia, pp. 552-557, Sep. 2004.

[2] M. R. Elliott, J. Drummond, and K. E. Barnard, "Subjective Appraisal of Infant Crying," Clinical Nursing Research, vol. 5, no. 2, pp. 237-250, May 1996.

[3] L. S. Franck et al., "Pain Assessment in Infants and Children," Pediatric Clinics of North America, vol. 47, no. 3, pp. 487-512, Jun. 2000.

[4] C.-Y. Chang et al., "Application of Incremental SVM Learning for Infant Cries Recognition," in IEEE 18th Int. Conf. on Network-Based Information Systems (NBiS), pp. 607-610, Sep. 2015.

[5] Y. Mima and K. Arakawa, "Cause Estimation of Younger Babies' Cries from the Frequency Analyses of the Voice," in IEEE Int. Symp. on Intelligent Signal Processing and Communications (ISPACS'06), pp. 29-32, Dec. 2006.

[6] R. R. Vempada et al., "Characterization of infant cries using spectral and prosodic features," in IEEE Nat. Conf. on Communications (NCC-2012), pp. 1-5, Feb. 2012.

[7] M. Petroni et al., "Classification of infant cry vocalizations using artificial neural networks (ANNs)," in IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP-95), vol. 5, pp. 3475-3478, May 1995.

[8] S. E. Barajas-Montiel and C. A. Reyes-Garcia, "Identifying Pain and Hunger in Infant Cry with Classifiers Ensembles," in IEEE Int. Conf. on Computational Intelligence for Modeling Control and Automation, and Int. Conf. on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06), vol. 2, pp. 770-775, Nov. 2005.

[9] A. Zabidi et al., "Classification of Infant Cries with Hypothyroidism Using Multilayer Perceptron Neural Network," in IEEE Int. Conf. on Signal and Image Processing Applications (ICSIPA), pp. 246-251, Nov. 2009.

[10] O. F. Reyes-Galaviz et al., "Evolutionary-Neural System to Classify Infant Cry Units for Pathologies Identification in Recently Born Babies." in 7th Mexican Inter. Conf. on Artificial Intelligent (MICAI'08), Atizapan de Zaragoza, Mexico, pp. 330-335, Oct. 2008.

[11] J. Saraswathy et al., "Infant Cry Classification: Time Frequency Analysis," in IEEE Int. Conf. on Control System, Computing and Engineering (ICCSCE), pp. 499-504, Nov. 2013.

[12] J. O. Garcia and C. A. Reyes-Garcia, "Mel-frequency cepstrum coefficients extraction from infant cry for classification of normal and pathological cry with feed-forward neural networks," in IEEE Proc. of the Int. Joint Conf. on Neural Networks, pp. 3140-3145, Jul. 2003.

[13] D. Lederman et al., "On the use of hidden Markov models in infants' cry classification," in IEEE 22nd Conv. of Electrical and Electronics Engineers in Israel, pp. 350-352, Dec. 2002.

[14] K. Santiago-Sanchez et al., "Type-2 fuzzy sets applied to pattern matching for the classification of cries of infants under neurological risk," in Emerging Intelligent Computing Technology and Applications, pp. 201-210, Springer, Sep. 2009.

[15] Q. Xie et al., "Automatic Assessment of Infants' Levels-of-Distress from the Cry Signals," IEEE Trans. on Speech and Audio Processing, vol. 4, no. 4, pp. 253-265, Jul. 1996.

[16] H.-Y. Yang and X.-X. Jing, "Performance Test of Parameters for Speaker Recognition System Based on SVM-VQ," in IEEE Int. Conf. on Machine Learning and Cybernetics, vol. 1, pp. 321-325, Jul. 2012.

[17] J. Garofolo et al., TIMIT Acoustic-Phonetic Continuous Speech Corpus LDC93S1. Web Download. Philadelphia: Linguistic Data Consortium, 1993.

[18] G. Lebrun et al., "Fast Pixel Classification by SVM Using Vector Quantization, Tabu Search and Hybrid Color Space," in Int. Conf. on Computer Analysis of Images and Patterns, pp. 685-692, Springer, Sep. 2005.

[19] T. Yu et al., "Combine Vector Quantization and Support Vector Machine for Imbalanced Datasets," in IFIP 19th World Congr. on Artificial Intelligence in Theory and Practice, pp. 81-88, Springer, Aug. 2006.

[20] J.A.K. Suykens et al., "Least Squares Support Vector Machines," in World Scientific, Singapore, Nov. 2002.

[21] J. Luts et al., "A Tutorial on Support Vector Machine-based Methods for Classification Problems in Chemometrics," Analytica Chimica Acta, vol. 665, issue 2, pp. 129-145, Apr. 2010.

[22] P. D. Welch, "The Use of Fast Fourier Transforms for the Estimation of Power Spectra: A Method Based on Time Averaging Over Short Modified Periodograms," IEEE Trans. on Audio and Electroacoustics, vol. 15, pp. 70-73, Jun. 1967.

[23] A. M. Gallo, “The Fifth Vital Sign: Implementation of the Neonatal Infant Pain Scale,” J. of Obstetric, Gynecologic, and Neonatal Nursing, vol. 32, issue 2, pp. 199-206, Mar. 2003.

[24] M. Goldstein, “k_n-nearest Neighbor Classification,” IEEE Trans. On Information Theory, vol. 18, issue 5, pp. 627-630, Jan. 1972.

[25] A. Liaw and M. Wiener, “Classification and Regression by randomForest,” R news, vol. 2/3, pp. 18-22, Dec. 2002.

[26] V. Svetnik, A. Liaw, C. Tong, J. C. Culberson, R. P. Sheridan, and B. P. Feuston, “Random Forest: A Classification and Regression Tool for Compound Classification and QSAR Modeling,” J. of chemical information and computer sciences, vol. 43, issue 6, pp. 1947-1958, Nov. 2003.

[27] L. Breiman, “Random Forests,” in Machine Learning, Springer, vol. 45, issue 1, pp. 5-32, Oct. 2001.

[28] A. Kuh, “Least Squares Kernel Methods and Applications,” in Soft Computing in Communications, Springer Berlin Heidelberg, vol. 136, pp. 365-387, 2004.

[29] L.-Z. Ding, and S. Liao, “Approximate Model Selection for Large Scale LSSVM,” in Asian Conf. on Machine Learning, Taoyuan, Taiwan, pp. 165-180, Nov. 2011.