Proceedings of the 2007 IEEE Symposium on Computational
Intelligence in Image and Signal Processing (CIISP 2007)

AUDIO ENVIRONMENT CLASSIFICATION FOR HEARING AIDS USING ARTIFICIAL NEURAL NETWORKS WITH WINDOWED INPUT

C. Freeman, R.D. Dony and S.M. Areibi

School of Engineering
University of Guelph
Guelph, ON, CANADA N1G 2W1
cfreeman,rdony,sareibi@uoguelph.ca

ABSTRACT

Excessive background noise is one of the most common complaints from hearing aid users. Background noise classification systems can be used in hearing aids to adjust the response based on the noise environment. This paper examines and compares two promising classification techniques, non-windowed artificial neural networks (ANN) and hidden Markov models (HMM), with an artificial neural network using windowed input. Results obtained show that an ANN with a windowed input gives an accuracy of up to 97.9%, which is more accurate than both the non-windowed ANN and the HMM. Overall, a windowed ANN is able to give excellent accuracy and reliability and is considered to be a good model for background noise classification in hearing aids.

1 Introduction

Patients with sensorineural hearing loss have a decreased ability to hear sounds at different frequencies. Hearing aids are often used to compensate for this loss. Patients who wear hearing aids, however, often complain of a number of different problems, most often difficulty hearing in environments with background noise (1). Most hearing aids simply amplify sounds in the required frequency range, hence both foreground and background noises are amplified. This makes hearing more difficult in noisy environments.

Because different noise environments have different frequency components, different acoustic environments require different filters to eliminate noise. Patients also tend to prefer different frequency responses from their hearing aids depending on the listening environment (2; 3). These types of programmable hearing aids are currently available on the market; however the user must change the setting manually (3). Automatic background noise classification systems can be used to automatically change the frequency program or to change the characteristics of a noise reduction or speech enhancement system.

A number of different studies have looked at the classification of audio environments, both for hearing aids and for other purposes, such as audio sample categorization for Internet applications. Two of the most promising techniques found for audio classification are artificial neural networks (ANN) and hidden Markov models (HMM) (4). This paper examines and compares these two background classification techniques and compares them to an artificial neural network that uses windowed input.

This paper extends the work by Buchler et. al. (4) by looking specifically at determining the type of background noise from a known background sample. It also proposes a novel approach, the windowed ANN, as a possible way to further increase the accuracy of the system.

The remainder of the paper is organized as follows: Section 2 gives some background on the classifiers and discusses the previous work on this topic, Section 3 discusses the methods used in this work, Section 4 presents and discusses the results and Section 5 presents the conclusions and future work.

2 Background

2.1 Artificial neural networks

Artificial neural networks are based on models of neurons and consist of layers of interconnected nodes. These nodes receive weighted input from previous layers and output signals to subsequent layers. The output is based on the weighted input, an internal threshold value and an activation function.

The network used is this work is a three-layer (one hidden layer) feed-forward perceptron with a variable number of hidden nodes. It is trained using back-propagation (5), which is a supervised learning technique that uses the error at the output to adjust the weights and thresholds in the model.

2.2 Hidden Markov models

Hidden Markov models are stochastic signal models, based on Markov chains (6). The model is represented as a number of discrete states, where the current state depends only on the last state and the current input. In a hidden Markov model the actual states are hidden and cannot be accessed directly. Instead, the model generates observable sequences that can be used to estimate the state of the system. The model is therefore extended to include observation probabilities within each of the states. In a discrete model, the number of observations is finite and the input values are quantized to fit the possible observation values.

An HMM can be represented fully by a matrix describing the state transition probabilities (A), a matrix describing the observation probabilities for each state (B), and a matrix de-

1-4244-0707-9/07/$25.00 ©2007 IEEE

scribing the initial state probabilities (π) (6). In this work, the model is trained using the Baum-Welch algorithm (6).

2.3 Previous work

A number of researchers have looked at audio classification using both ANNs and HMMs. A study by Nordqvist and Leijon (3) uses HMMs to classify three different audio environments. The work in (3) uses a two-stage classifier. The first classifier determines the environment while the second classifier picks out the individual speech and noise parts of the signal. The system was able to give classification accuracy from 96.7% to 99.5%, with false alarm rates from 0.2% to 1.7%. These results are very encouraging. However the number of environments tested was fairly limited. This study does show that the HMM is very promising as a possible classifier for audio environments.

Khan, Al-Khatib and Moinuddin (7) looked at classifying speech and music from unknown audio samples. They used a multi-layer perceptron feed-forward network, trained using back-propagation, and were able to attain a 96.6% accuracy. This is an impressive accuracy rate; however, there are a small number of possible classes used in this implementation. Similar results were achieved by Bugatti et. al. (8) using neural networks for the same problem. They achieved a total error rate of only 6% using the neural network, compared to the error rate of 17.7% using a Bayesian filter (8).

Buchler et. al. (4) studied HMMs and neural networks as well as several other classifiers for the purpose of sound environment classification. They used four separate sound classes consisting of noise, speech in noise, speech and music. The study found that the HMM was the most successful model for environment classification with an 88% accuracy rate, followed closely by the neural network with an 87% accuracy rate. These two models are very close, therefore it is difficult to definitively claim that the HMM is a better classifier.

From the above, it is clear that both the HMM and the ANN are suitable algorithms for audio classification.

3 Methods

3.1 Sound classes

The models are evaluated using four different classes of background noises: speech babble, traffic noise, typing and white noise. These categories are similar to the categories used by Nordqvist and Leijon in (3), with the addition of the percussive typing noise category identified by Kates in (2).

3.2 Features

The feature vector used in this work is similar to the feature vector described by Kates in (2). The feature vector consists of the mean frequency, the high and low frequency slopes and the envelope modulation of the sample. The mean frequency and the slopes are calculated on the log frequency scale [1] The features are calculated on a 200 ms frame, as required for the calculation of the envelope modulation. The regular neural network uses a single frame of the signal to determine the class, and the HMM uses a sequence of five frames, consisting of one second of audio input. The windowed neural network uses a variable window size of two to five frames.

Many previous works have tested many possible features for this application (7). The most appropriate feature vector is dependent both on the classes (9) and the classifier (4). At this phase of the study, the intent is simply to determine whether the windowed ANN is a reasonable classifier for this application, hence this relatively simple feature vector was selected for testing. Further work on this project will look at tailoring the feature vector to both the classifier and the classes chosen.

3.2.1 Mean frequency

The mean frequency the first moment of the of the log frequency spectrum. It gives a general description of the frequencies in which the majority of the signal is contained. To calculate the mean frequency, the sample is broken into its frequency components using a fast Fourier transform (FFT). The mean frequency (2) is then calculated as:

$$F_{mean} = \frac{\sum_{k=1}^{N/2} |F_k|}{\sum_{k=1}^{N/2} \frac{|F_k|}{f_k}} \tag{1}$$

where F_{mean} is the mean frequency as an FFT bin index, f_k is the frequency at index k, $|F_k|$ is the magnitude of the frequency response at index k, and N is the number of samples in the frame.

3.2.2 High and low frequency slopes

The high and low frequency components are separated by the mean frequency. The slopes of the high and low frequency components give a general description of the shape of the spectrum about the mean. The slope of both the low and high frequencies is determined by least-squares fit to the log frequency response (2). Both are given in the form:

$$y(k) = a_0 + a_1 log_2(k) \tag{2}$$

where a_0 is a constant, and a_1 gives the slope in dB/octave.

For the low frequency (2), parameters a_0 and a_1 are calculated as:

$$\begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = \begin{bmatrix} \sum_{k=1}^{L} \frac{1}{k} |F_k|_{db}) \\ \sum_{k=1}^{L} \frac{1}{k} |F_k|_{db} log_2 k \end{bmatrix} \tag{3}$$

where k is the FFT bin index, $|F_k|_{db}$ is the magnitude in dB of the FFT bin index, and L is the FFT bin index just below the mean frequency.

The high frequency slope is calculated in a similar manner, but using the frequencies above the mean.

3.2.3 Envelope modulation

The envelope modulation is the most complex of the four features in the vector. The envelope modulation attempts to describe how the frequency response changes over time. The process used to generate the envelope modulation is based on the process described in (2).

[1] Kates shows that this feature performed as well as the entire log frequency spectrum (2).

A short (200 ms) background noise segment is required to determine the envelope modulation. The segment is further divided into 6.4 ms segments, starting 3.2 ms apart, which gives a 50% overlap in the smaller segments. These numbers come directly from the Kates study (2), and are based on human auditory perception.

An FFT is taken for each segment. The average magnitude across all frequencies is calculated for each 6.4ms segment. The mean magnitude and the standard deviation are calculated across the time segments, and the envelope modulation is given as the mean magnitude over the standard deviation measured across the segments.

This is calculated as:

$$m_{env} = \frac{\mu}{\sigma} \tag{4}$$

where m_{env} is the envelope modulation, μ is the mean of the segment means, σ is the standard deviation of the segment means, and where μ is calculated as

$$\mu = \frac{1}{M}\sum_{i=1}^{M}\mu_{6.4i} \tag{5}$$

and

$$\sigma = \sqrt{\frac{1}{M}\sum_{i=1}^{M}(\mu 6.4i - \mu)} \tag{6}$$

where M is the number of 6.4 ms segments and $\mu_{6.4i}$ is the mean magnitude of the 6.4 ms segment i.

3.3 Data set

The data set consists of sound recordings of background environments, taken from the Freesound database at the Universitat Pompeu Fabra (10), and white noise generated in MATLAB. The files are separated into 1 second segments. Each class is taken from 37 1-second segments. For neural network, which only requires one frame of data per input, each 1-second segment was further separated into 4 250ms segments. From each 250ms segments, 1 200ms frame of features was extracted. The windowed ANN and the HMM each require more than one frame per input, hence each 1-second segment was used to produce a single set of 2 to 5 input vectors.

3.4 The non-windowed ANN

The network model used in this study is a three-layer (one hidden layer) feed-forward perceptron, trained by back-propagation. The input to the network is four nodes, with one node per feature. The output of the network is four nodes, with each node corresponding to a single class. The number of hidden nodes is variable. Since the initialization for the weights and thresholds is random, the results of 50 separate training runs were averaged to produce the results. The run with the best performance on the test set is also saved. The pseudocode for the implementation is presented in Fig. 1.

The network is tested using different numbers of hidden nodes (four to ten), and different numbers of training epochs (10,000 to 30,000).

```
for r=1:#_iterations{
     initialize new model
     for e=1:#_epochs{
          for i=1:#_training vectors{
               Generate forward terms
               Generate error terms
               Adjust weights
          }
     }
     test using training set
     add training set hit rate to avg
     test using test set
     add test set hit rate to avg
     if (test set hit rate > best){
          save network as best-run
     }
}
```

Fig. 1*:* Pseudocode for the ANN implementation

3.5 The hidden Markov model

The hidden Markov model implementation is slightly different than the neural network implementation, since a single model is required for each audio class. The set of four HMMs makes up a model set, and the class of the input vector is determined as the HMM with the highest probability in the set.

The state transition matrix (A) and the initial state probabilities (π) are initialized randomly. However, it has been shown that observation probability matrix (B) benefits from a better initial estimate (6), hence the observations are first clustered to give initial estimates for B, using K-means clustering. A discrete model is used, hence the values used in the B matrix are determined using a vector consisting of codebook values. The actual input values are quantized to their codebook values before they are input into the B matrix.

The HMMs are much more consistent since the initialization for the B matrix is not random. Therefore, the results are generated by averaging five HMMs. The model with the best accuracy on the test set is also saved. The pseudocode for this implementation is presented in Fig. 2.

The HMMs are trained using different number of classes (two to four) and different numbers of codebook values (three to six). The number of classes was restricted to four since beyond this number the initialization of the B matrix becomes difficult. The HMMs were also tested with different number of training iterations, from 10 to 30.

3.6 The windowed ANN

Although both ANNs and HMMs can match input patterns to output classes, HMMs have a theoretical advantage over ANNs as they can also track time-based changes. The ANN does not naturally contain a time-based component. We propose a new ANN method that uses a time-based input by modifying the input vector to be a windowed vector. In a windowed vector, the input to the neural network is modified to include past inputs by adding more nodes to the input layer, as seen in Fig. 3.

When the input is windowed, the size of the input window must be considered as an additional parameter. In this

```
for r=1:#_iterations{
      for m=1:#_classes{
            initialize new model
            for e=1:#_epochs{
                  for i=1:#_vectors{
                        forward-backward
                        Baum-Welch
                  }
            }
            add new HMM to HMM set
      }
      test using training set
      add training set hit rate to avg
      test using test set
      add test set hit rate to avg
      if (test set hit rate > best){
            save network as best-run
      }
}
```

Fig. 2*:* Pseudocode for the HMM implementation

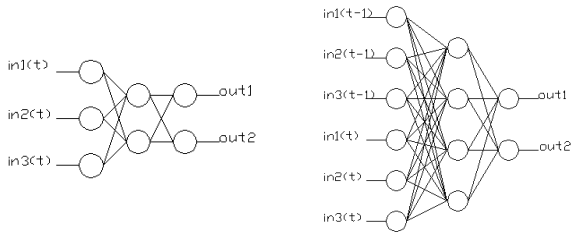

Fig. 3*:* ANNs using normal and windowed input vectors

case, the position of the sample in the window is not important since, in the vast majority of cases, the background is stationary. Therefore, the network is used to classify the background from the full sequence of inputs. In cases where the background is non-stationary, the change will propagate through the system, and should only cause a disturbance in the windows containing samples from more than one environment. For this system, the window is limited to five samples, which is only one second of data. Any disturbances could be smoothed using an averaging filter at the output.

4 Results and Discussion

4.1 The non-windowed ANN

The average accuracy of the neural networks on the training set was actually quite low, ranging from 75.5% to 78.8% accuracy on the testing set, depending on the number of hidden nodes and the number of training epochs. The best-run values were much higher than the average values, ranging from 84.9% to 92.7% accuracy on the testing set. This large range indicates that the artificial neural network is not a very reliable model. Neural networks are able to find local minima in the error space, but not a global minimum. Accordingly, it is possible to generate models that do not fully converge to a solution for the training set. This may be part of the reason for the relatively low average accuracy.

There is no definite trend with respect to the number of training epochs. A 20,000 epoch training period was selected for comparison, as it is the midrange of the tested values.

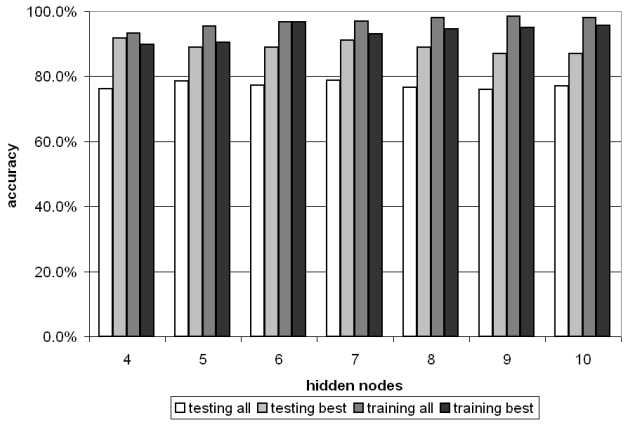

Fig. 4*:* Accuracy of neural networks using different number of hidden nodes

	selected				
actual	babble	traffic	typing	white	unknown
babble	87.5%	6.2%	0%	0%	6.2%
traffic	10.4%	75.0%	4.2%	0%	10.4%
typing	0%	4.2%	93.8%	0%	2.1%
white	0%	0%	0%	100%	0%

Table 1*:* Confusion matrix for the best-run of the 5-hidden node ANN

From the graph in Fig. 4, it can be seen that there are also very few definite trends with respect to the number of hidden nodes. There is a very slight decrease in both best-run and average accuracy for networks with more than seven hidden nodes. The network with five hidden nodes (trained with 20,000 epochs) was selected for comparison. For this model, the best-run accuracy for the testing set is 89.1%. Using these weights, the accuracy on the testing set was 90.5%. The average accuracy achieved is 78.6% for the testing set and 95.44% for the training set. This model is selected because it has fewer than seven nodes, and the smallest difference between the average and best-run accuracy, making the model slightly more reliable than the others. It also has the smallest difference between the best-run accuracy for the training and testing set, indicating that this model is more robust than the other models.

Practically, however, the number of hidden nodes does not greatly affect the final accuracy, and any relatively small number of hidden nodes would be an appropriate selection.

The confusion matrix for the testing set of the best run of the 5-hidden node matrix is presented in Table 1. The matrix shows that the majority of the confusion is between the babble and the traffic classes. This observation holds true for all network configurations for both the testing and the training sets. The neural network may require additional features to more fully separate these two classes.

4.2 The hidden Markov model

The results from the HMM tests show that the number of training iterations has little effect on the final accuracy of the model. The number of training iterations required for the HMM is much smaller than that required by the ANN. How-

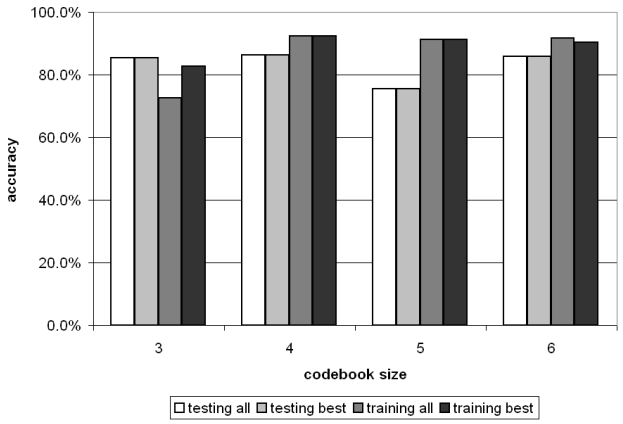

Fig. 5*:* Accuracy of the HMM using different codebook sizes for a 3-class HMM

ever, the training time is still fairly long as the matrices are much larger than the matrices found in the ANN. For a 4-input 4-output ANN with N hidden nodes, the network consists of a 4xN and an Nx4 weight matrix, and 4+N thresholds. For the HMM the number of possible vectors in the codebook vastly increases the size. An HMM with a three classes, a codebook size of Q and four inputs will have Q^4x3 B matrix, a 3x3 A matrix and a 3x1 π vector. Increasing the number of codebook values therefore increases the size of the HMM dramatically. Additionally, HMMs require one model per class, whereas ANNs require only one network.

HMMs are more reliable than ANNs. The majority of the cases tested had an average accuracy that was the same as the best-run accuracy, and those that were not the same differed only by the classification of one sample. This indicates that the models converged to a similar result every time they were trained. This is likely because the initialization for the HMMs is not random, but is based on a K-means clustering process.

There also seems to be little difference in accuracy between models with different numbers of classes, particularly when using a smaller number of codebook values. As the number of codebook values increases, the accuracy of both the 2-class and 4-class models decrease slightly. The 4-class models are more difficult to initialize with the K-means clustering, which may be an indication that there are not actually four distinct classes in all the environments being classified. Overall, the 3-class model appears to be the best model.

The parameter that has the greatest effect on the performance of the model is the number of codebook values. The results are relatively similar for codebooks with three, four and six quantized values. However, the performance of the codebook with five quantized values is quite a bit lower, as seen in Fig. 5. It is possible that the codebook with five values separates or combines a cluster of input vectors that would otherwise be classified differently. The 4-value codebook is selected for comparison because it is the most accurate for the 3-class model. It is also a relatively small codebook, making it faster to train and easier to fit in the relatively small memory found on DSP-based hearing aids.

The confusion matrix for the HMM implementation is

	selected			
actual	babble	traffic	typing	white
babble	54.5%	0%	45.4%	0%
traffic	9.1%	90.9%	0%	0%
typing	0%	0%	100%	0%
white	0%	0%	0%	100%

Table 2*:* Confusion matrix for the best-run set of the 3-class 4-codebook value HMM

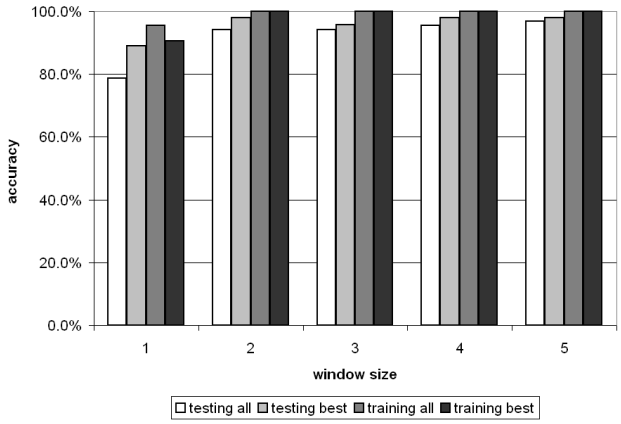

Fig. 6*:* Accuracy of the windowed ANN vs. window size

shown in Table 2. It is clear from Table 2 that the HMM has some difficulty classifying the babble sound class. However, the HMM most often confuses babble with typing. This is a further indication that different features may be needed to properly classify the babble class.

4.3 The windowed ANN

Testing with the non-windowed ANN showed that for the 4-input network, a network with five hidden nodes is a good model. Using the results from the non-windowed ANN tests, the windowed ANNs were set so that the ratio of hidden nodes to input nodes is 5:4. Window sizes from two to five samples were tested, and the results are presented in Fig. 6.

Results indicate that even a small window provides a significant increase in accuracy over the non-windowed ANN. The testing set best-run accuracies are between 95.8% and 97.9% depending on the window size, compared to 89.1% for the non-windowed ANN, and 86.4% for the HMM. Additionally, the average accuracy is increased significantly from just 78.6% for the ANN and 86.4% for the HMM to between 94.2% and 96.9%. The windowed ANN is clearly more accurate than both the non-windowed ANN and the HMM.

The difference between the average and best-run accuracy for the non-windowed ANN is 10.5%. The windowed ANNs show a difference of only 1.0% to 3.7%. This indicates that the windowed ANN will produce more consistent results and therefore is easier to train.

One of the major benefits of the windowed ANN is that it is robust. Although the non-windowed ANN is able to generalize in certain cases, the average accuracies for the training and testing sets are quite different. For the non-windowed ANN, the difference between the average accuracy for the

	selected				
actual	babble	traffic	typing	white	unknown
babble	100%	0%	0%	0%	0%
traffic	8.3%	91.7%	0%	0%	0%
typing	0%	0%	100%	0%	0%
white	0%	0%	0%	100%	0%

Table 3*:* Confusion matrix for the 2-sample windowed ANN

	Testing set		Training set	
Classifier	average	best-run*	average	best-run*
ANN	78.6%	89.1%	95.4%	90.5%
HMM	92.3%	92.3%	86.4%	86.4%
WANN	94.2%	97.9%	100.0%	100.0%

Table 4*:* Accuracy of three classifiers. *Note that best-run indicates the run giving the highest accuracy on the testing set.

training and testing sets is 16.9%. For the windowed ANN, this difference is reduced to 5.8% for the 2-sample window, and just 3.1% for the 5-sample window.

The average accuracy increases slightly as the window size increases. However, the best-run accuracy is relatively flat, with the 2, 4 and 5-sample windows having a best-run accuracies of 97.9%, and the 3-sample window having a slightly lower best-run accuracy at 95.8%. The reliability also increases slightly as the window size increases. The larger window size is therefore likely easier to train, but the smaller window sizes are also capable of achieving excellent accuracy.

Although the larger windows are slightly more accurate on average, the 2-sample window was selected for comparison since it is smaller and is therefore faster to train. It also has a smaller computational load, and easier to store on the relatively small memory found in a hearing aid.

The confusion matrix for a the ANN using window size of two samples is presented in Table 3. There is only one misclassified sample, which is a traffic sample that is misidentified as babble. This is similar to the problems seen in the non-windowed ANN.

4.4 Comparison of the three classifiers

The final accuracy of all three systems is presented in Table 4. The windowed ANN has the best accuracy for both the average and the best-run models. It is also the most general model. The HMM is the most reliable model; however the windowed ANN is more reliable than the non-windowed ANN.

5 Conclusions

This paper examined and compared two background classification techniques, non-windowed artificial neural networks and hidden Markov models, and compared these two techniques to an artificial neural network using windowed input.

The results show that the windowed ANN is more accurate than both the HMM and the non-windowed ANN. Although the HMM is still the most reliable model, the windowed ANN is more reliable than the non-windowed ANN, and is also a more general model. In addition to being slightly more accurate than the HMM, the windowed ANN is smaller and takes less time to train. This size difference is important as the system intended for implementation in a hearing aid, which has limited space and computational power. Overall, using an ANN with a windowed input appears to be an excellent choice for background classification in a hearing aid.

Future work will look at determining more appropriate sound classes, and determining more appropriate features to further refine the classification ability of the model.

References

[1] R. Plomp, "Auditory handicap of hearing impairment and the limited benefit of hearing aids," *Journal of the Acoustical Society of America*, vol. 63, no. 2, pp. 533 – 549, 1978.

[2] J. Kates, "Classification of background noises for hearing-aid applications," *Journal of the Acoustical Society of America*, vol. 97, no. 1, pp. 461 – 470, 1995.

[3] P. Nordqvist and A. Leijon, "An efficient robust sound classification algorithm for hearing aids," *Journal of the Acoustical Society of America*, vol. 115, no. 6, pp. 3033 – 3041, 2004.

[4] M. Buchler, S. Allegro, S. Launer, and N. Dillier, "Sound classification in hearing aids inspired by auditory scene analysis," *EURASIP Journal on Applied Signal Processing*, vol. 2005, no. 18, pp. 2991 – 3002, 2005.

[5] S. S. Haykin, *Neural networks: a comprehensive foundation*. New York: Macmillan, 1999.

[6] L. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257 – 286, 1989.

[7] M. K. S. Khan, W. G. Al-Khatib, and M. Moinuddin, "Automatic classification of speech and music using neural networks," *MMDB 2004: Proceedings of the Second ACM International Workshop on Multimedia Databases*, pp. 94 – 99, 2004.

[8] A. Bugatti, A. Flammini, and P. Migliorati, "Audio classification in speech and music: A comparison between a statistical and a neural approach," *Applied Signal Processing*, vol. 2002, no. 4, pp. 372 – 378, 2002.

[9] D. Zongker and A. Jain, "Algorithms for feature selection: An evaluation," *Proceedings of the 13th International Conference on Pattern Recognition*, vol. 2, pp. 18 – 22, 1996.

[10] U. P. F. Institut Universitari de L'audiovisual, "The freesound project." [Online]. Available: http://freesound.iua.upf.edu/