Feature Normalization in SVM Speaker Verification using Telephone Speech

Thembisile Mazibuko (*thembi@crg.ee.uct.ac.za*), Daniel Mashao (*daniel.mashao@sita.co.za*)

Speech Technology and Research
Communication Research Group
University of Cape Town

***Abstract* — Histogram Equalization, traditionally an image processing technique, is applied in a Support Vector Machine based text-independent speaker verification system using telephone speech from the NIST 2000 Speaker Recognition Evaluation. The performance achieved by applying HEQ is compared to that of common linear feature normalization techniques Cepstral Mean Subtraction and Mean Variance Normalization. The results show an improvement in SVM classification performance with HEQ achieving the best performance. Feature normalization also results in a reduction of the average SVM training time and the required resources for storing the SVM speaker models.**

***Index Terms*— speaker verification, support vector machine, histogram equalization**

I. Introduction

Speaker verification refers to the task of using computers to authenticate the identity of a user, using speech as a form of identification. There are two categories of speaker verification: *text-dependent* and *text-independent*. The focus of this research is the latter. In text-dependent speaker verification the user is required to use a predetermined phrase in order to access the system. These systems generally give good results but have limited flexibility as the user is required to remember the specific phrase required to gain entry to the system. The alternative text-independent variety of speaker verification is more flexible but is also a significantly more challenging task.

Speaker verification systems generally operate in two modes; training and testing [1]. The training mode typically comprises of these components: *data acquisition*, *feature analysis* and *speaker modelling* while the testing mode replaces the speaker modelling component with a *decision-making/classification* component. The data-acquisition and feature analysis modules are together referred to as the *front-end* of the system while the speaker modelling and decision-making components are collectively the system *back-end*.

The development and expansion of the telephone network has meant that speech communication and speech-based technologies such as speaker verification can be used over long distances. Telephonic shopping or banking are examples of applications in which the ability to remotely verify a users identity would be valuable. While research has shown that speaker verification systems tend to work very well in controlled laboratory conditions; performing as well as, and sometimes better than human listeners [2] these systems tend to suffer severe degradation in performance when the conditions of application are less than ideal. Noise and mismatch in training and testing conditions and channel effects are major contributors to creating these imperfect conditions.

Noise refers to background noise caused by traffic or people conversing in the background, distortions to the speech signal caused by transmission over telephone lines and ambient noise. The mismatch between training and testing conditions can be related to either the user (e.g. change in the speaker's physical or emotional state i.e. health or mood) or the environment. The environmental factors include collecting the speech data in different environments (e.g. a noisy office or a quiet room), using different transmission media (cellular phone and landline) or different equipment (handset or microphone types). The speech signal must be transmitted through some channel prior to reaching the recording device. Transmission over the channels such as the telephone channel introduces some distortion to the original signal [3]. The additive environmental noise $n(t)$ and linear filtering effects from communication channels are often modelled as [4]:

$$s_r(t) = \left[s_o(t) + n(t)\right] \otimes h(t) \qquad (1)$$

where $s_o(t)$ and $s_r(t)$are the original and recorded speech signals, respectively. The sum of the original speech signal and noise is passed through a linear time-invariant filter $h(t)$ which represents the linear filtering effects of the channel as well any distortion that might occur due to mismatched training and testing conditions.

Normalization techniques have been used for a long time in speech processing to compensate for the varying effects of

noise, mismatch and channel distortions. The aim of feature normalization is to make the features more robust in adverse conditions. In this research, the performance of two linear feature normalization techniques, Cepstral Mean Normalization (CMN) and Mean Variance Normalization (MVN) which are widely used in speech processing, is compared to that of Histogram Equalization (HEQ) in a text-independent speaker verification task using telephone speech. HEQ is traditionally an image processing technique which has recently been successfully applied to speech processing [4-6]. In [4] HEQ was found to significantly outperform CMN and MVN when applied to a similar task using the popular Gaussian Mixture Model (GMM). In this research, these normalization techniques are applied in a Support Vector Machine (SVM) [7] based speaker verification system.

The SVM is a supervised machine learning technique that learns the decision surface through a process of discrimination and is characterized by good generalization [7, 8]. It has been successfully applied to several pattern recognition tasks including tasks in speech processing [9-11] .

The remainder of this paper is organized as follows; the next session discusses the feature normalization techniques which are applied in this work in more detail. Section III provides an introduction to SVM classification. The experimental procedure used is reviewed in section IV. A discussion of the results and conclusions then follows in section V.

II. Feature Normalization

Feature-based normalization techniques aim to make the features which are generated during the parameterization process more robust to mismatched conditions [4]. Cepstral distributions for clean data are generally well behaved and approximately normal but take on a significantly different profile in the presence of noise [12]. Feature normalization techniques aim to normalize the cepstral distributions in the presence of noise.

A. Cepstral Mean Normalization

Cepstral Mean Normalization (CMN) or Cepstral Mean Subtraction (CMS) aims to compensate for the effect of the telephone channel on the speech by subtracting the long-term average from the cepstral coefficients on a per-utterance basis [13]. This technique works under the assumptions that the linear filtering effect which telephone transmission tends to have on the speech signal is time invariant and that the duration of the speech signal in question is long enough such that the average speech spectrum is flat [4]. CMN has the effect of causing the mean of the distribution of the compensated feature values to be equal to zero which normalizes the first moment of the distribution and removes time-invariant distortions from the signal [4, 14].

Being a linear normalization technique CMN is only able to compensate for the linear effects of additive noise and telephone transmission and is unable to normalize nonlinear distortions on the speech signal. The application of CMN only affects the first moment of the data distribution. Research has also found a tendency of CMN to degrade system performance when applied to clean data [15].

B. Mean Variance Normalization

Mean Variance Normalization is a natural extension of CMN which normalizes the first two moments of the data distribution by applying the transformation [16]:

$$x' = \frac{x - \mu}{\sigma} \tag{2}$$

where μ is the long term mean and σ is the standard deviation.

The effect of applying MVN is the normalization of the long-term mean to zero and the data variance to unity. Like CMN, MVN is only able to compensate for linear distortions to the data.

C. Histogram Equalization

As mentioned in the preceding sections, CMN and MVN which are linear feature compensation techniques have the limitation of being able to normalize only the first two moments (i.e. mean and variance) in the feature distribution. Histogram Equalization (HEQ) is traditionally an image processing feature compensation technique which is able to compensate for the first two moments as well as higher order moments (e.g. skew) [4]. The basis of HEQ is that the distribution of the cepstral coefficients in the test data should be identical to that of the training data [16]. That is, HEQ applies a non-linear transformation to data which has one probability distribution and matches it to another, reference distribution.

The aim is therefore to provide a transformation $x(y)$ which converts the probability distribution $p_y(y)$ of the noisy speech into a reference distribution $p_x(x)$ corresponding to the clean speech [17]. That is, for a random variable y with probability density function $p_y(y)$, a function $x = F(y)$ which maps $p_y(y)$ into a reference distribution $p_x(x)$ can be obtained by equating the cumulative distribution functions (CDF) of x and y such that [17]:

$$C_y(y) = C_x(x) = C_x(F(y)) \tag{3}$$

The transformation function $F(y)$ can then be obtained from the cumulative histogram of the noisy speech and the reference cumulative histogram for the clean speech so [17]:

$$x(y) = F(y) = C_x^{-1}\left[C_y(y)\right] \tag{4}$$

In general, *F(y)* is monotonic non-decreasing and nonlinear and, under the assumption of statistical independence, HEQ is applied to each cepstral coefficient independently. Figure 1 shows the histograms of clean and noisy speech data feature vector taken from the same speaker, before and after applying HEQ, respectively. As can be seen, applying HEQ brings the histogram for the noisy speech closer to matching the clean data histogram.

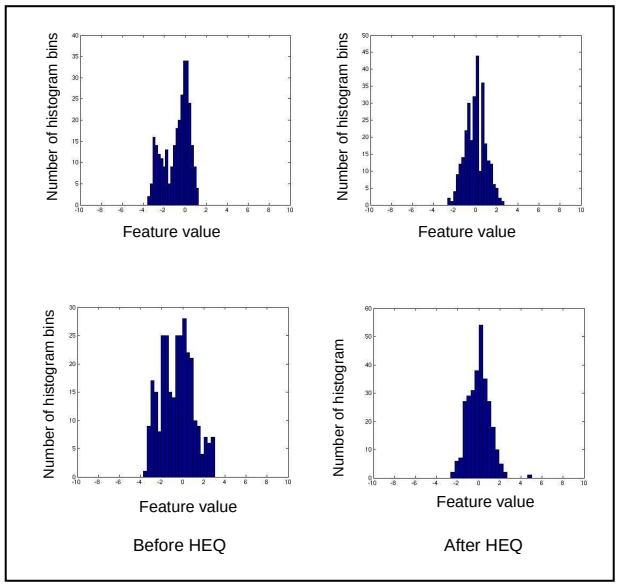

Figure 1: Feature vector histogram for clean and noisy data before and after HEQ.

The following section gives a brief introduction to classification using the Support Vector Machine.

III. Support Vector Machine Classification

The idea behind the formulation of the SVM can be expressed as follows [18]: the input vectors **x** are mapped onto high-dimensional space ***H*** (typically referred to as the *feature space*) through some nonlinear mapping,

$$\Phi : \mathbf{R}^N \rightarrow \mathbf{H} \qquad (5)$$

Thus the SVM performs binary pattern classification by implicitly mapping the training data onto a higher dimensional feature space [19].

A. The Linearly Separable Case

The simplest case in SVM classification is that of linear machines trained on separable data. Given training data $\{\mathbf{x}_i, y_i\}; \mathbf{x}_i \in R^N$ and $y_i \in \{-1,+1\}$ suppose there is a hyperplane which separates the data classes. There is an infinite number of decision boundaries that could be chosen for separating the data. However, among all hyperplanes separating data, there exists a unique one that results in the maximum margin of separation between the data classes. This optimal separating hyperplane (OSH) is orthogonal to the shortest line connecting the convex hulls[1] of the two classes, and intersects this line halfway between the classes [18].

The training data vectors which lie on the edges of the margin are referred to as the *support vectors*. These vectors are the critically important points in the training data. If training were repeated with all other training points removed or rearranged (but not so as to cross the margin-edge hyperplanes), the same separating hyperplane would be found [8].

In order to be applicable to real classification problems the SVM formulation must be extendable to nonlinearly separable data. This scenario is discussed below.

B. The Nonlinearly Separable Case

To extend the SVM algorithm to data that are not linearly separable, *slack variables* are introduced and the algorithm is "allowed" to misclassify some of the vectors during training, at a cost C. A higher C value denotes a greater penalty for misclassified training data.

As shown in figure 2, the misclassified vectors as well as those which lie within the region of the margin are the support vectors. The cost of error associated with the misclassification of training data is a user-defined parameter.

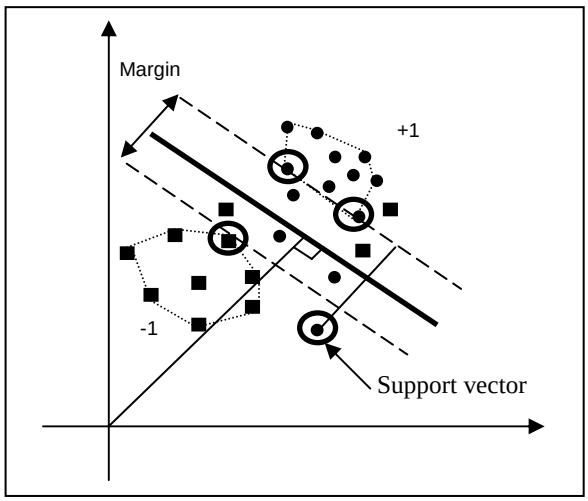

Figure 2: Classifying nonlinearly separable data with SVM.

C. Defining Nonlinear Decision Boundaries

In order to classify data requiring nonlinear decision boundaries, the SVM uses the *Kernel trick*. Recall that in support vector classification the input vectors **x** are mapped into a high-dimensional feature space ***H*** through some nonlinear mapping. This mapping results in a nonlinear decision boundary in the input space. Nonlinear SVM

[1] A convex hull of a set of points is the smallest convex set that includes all the points.

classification works on the premise that for constructing the separating hyperplane in the feature space ***H***, the space defined by ***H*** does not need to be considered in explicit form, we only need to be able to calculate the inner products between the support vectors and the vectors in the feature space [7, 20]. The SVM algorithm thus depends only on the dot product and if we define a function ***K***, referred to as the *kernel function*, such that

$$\mathbf{K}(\mathbf{x}_i \cdot \mathbf{x}_j) = \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j), \quad (6)$$

then by substituting the kernel function for the dot-product the algorithm does not need the explicit value of the dot product. In order to qualify the function used for the kernel must satisfy a set of restrictions called *Mercer's conditions*. Table 1 shows some examples of popular kernel functions.

Table 1: Common kernel functions

Polynomial	$\mathbf{K}(\mathbf{x},\mathbf{y}) = (\mathbf{x} \cdot \mathbf{y} + 1)^p$
Gaussian Radial Basis Function (GRBF)	$\mathbf{K}(\mathbf{x},\mathbf{y}) = e^{-\|\mathbf{x}-\mathbf{y}\|^2/2\sigma^2}$
Sigmodial Function	$\mathbf{K}(\mathbf{x},\mathbf{y}) = \tanh(\kappa\mathbf{x} \cdot \mathbf{y} - \delta)$

Using kernels that satisfy Mercer's condition, a nonlinear decision function can be created in the input space which is equivalent to a linear decision function in the feature space.

Figure 3 shows a simplified graphical representation of this principle applied to a two-dimensional problem. As can be seen the data requiring classification, which are nonlinearly separable in the input space ***X***, become linearly separable in the high dimensional feature space ***H***.

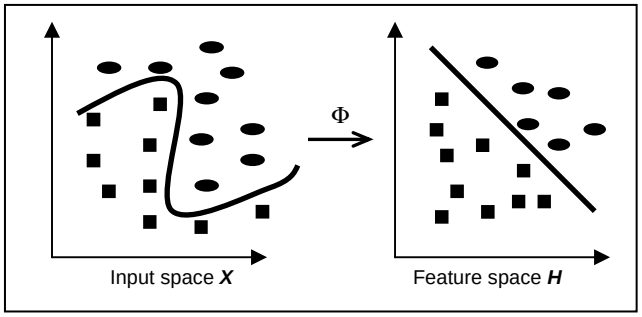

Figure 3: Nonlinear decision boundaries in SVM.

D. SVM Limitations

While the SVM is characterized by good generalization performance, a major limitation for this classifier is its speed and size, primarily the extremely long training time [8]. This limits not only the applicability of SVM, particularly in applications where real-time processing is required, but also the research done with the SVM as results take a long time to achieve. The SVM kernel also has parameters which require tuning. An example of this is the σ value in the Gaussian Radial Basis Function kernel. This is an important issue in optimizing SVM performance and is a valid research topic on its own, as is the best choice of kernel for a particular problem [21-23].

IV. Experimental Setup

The SVMTorch [24] package was used to build the SVM classification engine used in this research. A GRBF kernel was used. As mentioned in the previous section, finding optimum parameters for the SVM kernel can be a challenging and time consuming task. Since the aim of this research was to determine the effect of feature normalization on the performance of the SVM and, due to time restrictions, optimizing the kernel parameters was beyond the scope of this work. It is assumed that the effects of feature normalization on this non-optimal SVM speaker verification system would be similar to the effects on the performance of an optimized SVM classifier.

A. Experimental Corpus

The experiments conducted for this research used a subset of the NIST[2] 2000 Speaker Recognition Evaluation (SRE) task [25]. It was not possible to perform the full task, which consists of over 4000 trials because of the long training time associated with the SVM classifier. Two scenarios were considered. In the first case the training and testing handset types were matched i.e. both carbon of both electret. In the second case the training and testing handset types were mismatched i.e. training with an electret handset and testing with a carbon set. The NIST 2000 SRE task is configured such that even when the handset types matched, the speaker always used different handsets.

Each trial in the NIST 2000 SRE tasks consists of ten impostor trials and one target trial for each test file. Each test file is therefore involved in eleven trials. For this research twenty test cases were chosen for the matching handset type experiments and another twenty for the mismatched handset type resulting in a total of 200 impostor trials and twenty target trials for each scenario.

B. Methodology

A 17-dimensional feature vector using Mel-Frequency Cepstral Coefficients (MFCC) was used. Pre-emphasis filtering was applied with the filter coefficient value set to 0.97. A filterbank order of 26 was used for the MFCC feature set. These pre-processing procedures as well as the application of a Hamming window were conducted using the Edinburgh Speech Tools toolbox [26].
A simple energy-based voice activity detection scheme based was applied which removed 30-40% of the speech frames. Only the stationary features were used and thus the dynamic

[2] National Institute of Standards and Technology

delta and delta-delta features were not used. In [27] it is recommended that the feature attributes be scaled prior to applying the SVM in order to avoid numerical problems and prevent attributes in greater numerical ranges dominating the result. In [28] the author reported an improve performance in an SVM-based speaker identification task when scaling was performed by applying the following equation

$$x_n = \frac{x}{\max(abs(\mathbf{x}))} \quad (7)$$

such that x_n is in the range [-1, +1]. This scaling was applied to the feature attributes in this research before applying SVM.

A background model was built using speech data from the NIST 1999 corpus. A different corpus from that used to generate the results was used in order to avoid introducing a bias to the system. The following section discusses the results which were achieved in this work.

C. *Measuring System Performance*

The performance of the speaker verification system is reported in terms of the Equal Error Rate (EER). There are two types of error that can occur in verification systems; False Acceptance Error (FAR) where an impostor in wrongfully authenticated and False Rejection Error (FRR) where an valid or authentic user of the system is refused entry to the system. The EER is the point where the FRR and FAR are equal and is typically expressed as a percentage.

V. Results and Conclusions

The results, as seen in table 2, show that the linear feature normalization techniques, CMN and MVN, have the same effect on the SVM classifier performance. In the matched handset test the application of these led to degradation in system performance with the EER increasing by 4.5% from the 30% achieved by the baseline. This is in line with the finding in [15] that CMN tends to degrade system performance where there is no mismatch. HEQ performed as well as the baseline system in this case.

The similarity of the CMN and MVN results can be attributed to the scaling of the feature attributes which was performed in these experiments. Scaling the attributes effectively normalizes the variance of the feature vectors and since both CMN and MVN normalise the long-term mean this results in similar feature vectors and thus similar results.

Feature Normalization Type	Matched Handset EER [%]	Mismatched Handset EER [%]	Average Size of Model [KB]	Average No. Support Vectors in Model
Baseline	30.00	50.00	1945.6	10549.05
Cepstral Mean Normalization	34.50	45.50	419.80	3309.15
Mean Variance Normalization	34.50	45.50	419.80	3309.15
Histogram Equalization	30.00	44.50	320.64	2026.45

Table 2: Effect of feature normalization on SVM performance

In the second scenario where mismatched handset types were used for training and testing all the feature normalization techniques improved the system performance with the application of HEQ leading to the greatest performance.

In the case of the HEQ and baseline systems we were also able to compare the average training times for each model as these systems were built on computers with matching specifications. The computer models used were Intel Pentium 4 with a 3.20GHz central processing unit. It was found that the average training time for the baseline was 20179.72 ± 9186.14 seconds while the application of HEQ reduced this to 6087.01 ± 4534.79 seconds, almost a 70% reduction in average training time. The time was measured using the Linux/Unix *time* command.

The resources required for speaker models were also taken note of. As noted in Table 2, applying HEQ allows the SVM classifier to find a smaller SVM solution requiring less support vectors to define each model and also reduced the amount of storage space required for the speaker models from an average 1945.60 KB to 320.64 KB per model. These results echo Wan's assertion in [11] that larger SVM solution are not only less sufficient but also require more storage space as well as more computation.

VI. Future Work

Future research in this regard would include optimizing the SVM parameters in order to allow the results to be compared to the performance of state-of -the-art speaker verification systems. Also the effect of HEQ on the performance of other SVM kernels should be investigated.

REFERENCES

[1] J. P. Campbell, "Speaker Recognition: A Tutorial," *Proceedings of the IEEE*, vol. 85, pp. 1437-1462, 1997.

[2] A. Schmidt-Nielsen and T. H. Crystal, "Speaker Verification by Human Listeners: Experiments Comparing Human and Machine Performance Using the NIST 1998 Speaker Evaluation Data," *Digital Signal Processing*, vol. 10, pp. 249-266, 2000.

[3] S. Molau, "Normalization in the Acoustic Feature Feature Space for Improved Speech Recognition," in *Faculty of Mathematics, Information and Natural Sciences*. Aachen: RWTH, 2003.

[4] M. Skosan, "Histogram Equalization for Robust Text-Independent Speaker Verification in Telephone Environments," in *Department of Electrical Engineering*: University of Cape Town, 2005.

[5] J. C. Segura, C. Benitez, A. d. l. Torre, A. J. Rubio, and J. Ramirez, "Cepstral Domain Segmental Nonlinear Feature Transformations for Robust Speech Recognition," *IEEE Signal Processing Letters*, vol. 11, pp. 517-520, 2004.

[6] A. d. l. Torre, J. C. Segura, A. M. Peinado, A. J. Rubio, J. L. Perez-Corrdoba, and M. C. Benitez, "Histogram Equalization of Speech Representation for Robust Speech Recognition," *IEEE Transactions on Speech and Audio Processing*, vol. 13, pp. 355-366, 2005.

[7] V. Vapnik, *The Nature of Statistical Learning*, Second Edition ed: Springer, 1999.

[8] C. J. C. Burges, "A Tutorial on Support Vector Machines for Pattern Recognition," *Data Mining and Knowledge Discovery*, vol. 2, pp. 121-167, 1998.

[9] K. K. Chin, "Support Vector Machines Applied to Speech Pattern Classification," in *Computer Speech and Language Processing, Engineering Department*: University of Cambridge, 1999.

[10] A. Ganapathiraju, "Support Vector Machines for Speech Recognition," in *Department of Electrical and Computer Engineering*: Mississippi State University, 2002.

[11] V. Wan, "Speaker Verification Using Support Vector Machines," in *Department of Computer Science*: University of Sheffield, 2003.

[12] J. P. Openshaw and J. S. Mason, "On the Limitations of Cepstral Features in Noise," presented at International Conference on Acoustics, Speech and Signal Processing (ICASSP), Adelaide, Australia, 1994.

[13] S. v. Vuuren, "Comparison of Text-Independent Speaker Recognition Methods on Telephone Speech with Acoustic Mismatch," presented at International Conference on Spoken Language Processing (ICSLP), Philadelphia, USA, 1996.

[14] B. S. Atal, "Effectiveness of Linear Prediction Characteristics of the Speech Wave for Automatic Speaker Identification and Verification," *Journal of the Acoustical Society of America*, vol. 55, pp. 1304-1312, 1974.

[15] R. J. Mammone, X. Zhang, and R. P. Ramachandran, "Robust Speaker Recognition-A Feature Based Approach," in *IEEE Signal Processing*, vol. 13, 1996, pp. 58-71.

[16] Y. Obuchi and R. M. Stern, "Normalization of Time Derivative Parameters using Histogram Equalization," presented at European Conference on Speech Communication and Technology (Eurospeech), Geneva, Switzerland, 2003.

[17] A. d. l. Torre, J. C. Segura, A. M. Peinado, and A. J. Rubio, "Non-linear Transformations of the Feature Space for Robust Speech Recognition," presented at International Conference on Acoustics, Speech and Signal Processing (ICASSP), Orlando, Florida, USA, 2002.

[18] B. Schölkopf, C. J. C. Burges, and A. J. Smola, *Advances in Kernel Methods-Support Vector Learning*: MIT Press, 1999.

[19] A. Shilton, "Design and Training of Support Vector Machines," in *Department of Engineering*: University of Melbourne, 2006.

[20] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A Training Algorithm for Optimal Margin Classifiers," presented at Workshop on Computational Learning Theory, Pittsburgh,PA, 1992.

[21] O. Chapelle, V. Vapnik, O. Bousquet, and S. Mukherjee, "Choosing Multiple Parameters for Support Vector Machines," *Machine Learning*, vol. 46, pp. 131-159, 2002.

[22] K. Duan, S. S. Keerthi, and A. N. Poo, "Evaluation of Simple Performance Measures for Tuning SVM Hyperparameters," *Neurocomputing*, vol. 51, pp. 41-59, 2003.

[23] P. Erasto, "Support Vector Machines-Backgrounds and Practice," Rolf Nebanlinna Institute, 2001.

[24] R. Collobert and S. Bengio, "SVMTorch: Support Vector Machines for Large-Scale Regression Problems," *Journal of Machine Learning Research*, vol. 1, pp. 143-160, 2001.

[25] "The 2000 NIST Speaker Recognition Evaluation Plan, Version 1.0," vol. 2006: http://www.nist.gov/speech/tests/spk/2000/doc/spk-2000-plan-v1.0.htm.

[26] "The Edinburgh Speech Tools Library," http://www.cstr.ed.ac.uk/projects/speech_tools/.

[27] C.-W. Hsu, C.-C. Chang, and C.-J. Lin, "A Practical Guide to Support Vector Classification."

[28] R. Jhumka, "Evaluation of Different Support Vector Machine Kernels," in *Department of Electrical Engineering*: University of Cape Town, 2004.

Thembisile Mazibuko is in her second year of study towards an MSc Electrical Engineering qualification at the University of Cape Town under the supervision of Dr. Daniel Mashao.