On the Effect of the Implementation of Human Auditory Systems on $q$-Log-Based Features for Robustness of Speech Recognition Against Noise

HILMAN F. PARDEDE, ASRI R. YULIANI, AND AGUS SUBEKTI

Research Center for Informatics
Indonesian Institute of Sciences
Jl. Cisitu No 21/154D Bandung, INDONESIA
E-mail: {hilm001,asri006,agus075}@lipi.go.id

Mimicking human auditory systems as well as applying mean normalization in feature extraction are widely believed to improve the robustness of speech recognition. Traditionally, the normalization is conducted in the log domain by subtracting the features with their long-term mean. Some studies have found that the use of power functions instead of log yield more robust features. In previous studies, a $q$-logarithmic function ($q$-log), which is also a power function, was used to derive a normalization method. The method, called $q$-mean normalization ($q$-MN) in this paper, was found more effective than conventional normalization methods. In these works, $q$-MN was still applied in the power spectral domain. Here, the method is applied after mapping the power spectra on human auditory systems, and, after an analysis on the effect of the method on noisy speech, we propose a blind and adaptive normalization technique to determine a suitable $q$ in $q$-MN. The experiments show that the proposed features are more robust than conventional features such as MFCC. The results also confirm that using nonlinear resolutions inspired by human auditory systems benefits speech recognition and is better than using a uniform resolution.

Keywords: $q$-logarithm, robust speech recognition, feature normalization, human auditory, adaptive

1. Introduction

In the past decades, automatic speech recognition (ASR) technologies have been explored in many studies. However, their practical applications are still very limited due to their unreliable performances in real environments where background noise, reverberation, and competing speakers exist [1, 2]. Environmental noise is one of major causes that degrades the performance of ASR and numerous methods for improving the robustness of ASR in noisy environments have been proposed over the last decades. These methods are applied in the front-end, i.e. feature extraction process, and/or the back-end, i.e. the acoustic models. Methods for the front-end include noise removal methods, such as spectral subtraction [3] and vector Taylor series (VTS) [4, 5], and features that are robust against noise, such as Power normalized cepstral coefficients (PNCC) [6], normalized modulation cepstral coefficient (NMCC) [7], and frequency domain linear prediction (FDLP) [8]. The methods for the back-end aim at adapting acoustic models, which are trained in quiet environments, into noisy environments. This is usually done by retraining the model with noise-corrupted speech [9], adapting the model using parallel model combination (PMC) [10] and/or VTS model adaptation [11, 12].
Mel Frequency Cepstral Coefficients (MFCC) is arguably the most commonly used feature for ASR. MFCC is obtained by taking the log of mel-weighted spectra to separate the envelope of the spectra, which carries the information about the content of speech, from the speech signals. However, while MFCC can achieve satisfactory performances in controlled conditions, i.e. in noise-free conditions and for read speech, it is not robust against noise. It has been argued that the use of the log function is one of the reasons for it to be sensitive to noise [13]. The log function has large dynamics for values between 0 and 1 making it sensitive to the changes in the low energy regions of speech, which may contain important information about speech. When these regions are distorted by noise, there will be a large mismatch between the corresponding MFCC and the MFCC of clean speech which causes significant performance drops for ASR. Many studies have used alternative functions such as power functions instead of log in the extraction process. They are chosen because they have less dynamics in low energy regions, making them less sensitive to noise. The examples are Perceptual linear prediction (PLP) [14], PNCC, [6], and \( q \)-log normalized cepstral coefficients (QLNCC) [15]. However, other studies argue that mapping of the spectra using the root functions does not necessarily improve the speech recognition performance and their effectiveness are achieved when they are normalized [16, 17]. Normalizing the root-based features empirically appears to have minimize additive distortions [16, 18, 19]. Conventional normalizing approach is simply subtracting the features with its long term average [17]. However, root functions do not share the same properties as the log function. Therefore, their feature normalization methods should be modified accordingly.

In previous studies, the use of the \( q \)-log function for feature extraction has been investigated [19, 15]. This function, which is a of power function, is widely used in non-extensive statistics [20] to explain non-extensive phenomenon in many complex systems. In [19], \( q \)-log and its properties are utilized to derive a normalisation method. The method was proven to be more effective than conventional normalization methods for dealing with additive and convolutional noises. The method, called \( q \)-log spectral mean normalization (\( q \)-LSMN), is applied in the power spectral domain and \( q \)-log is used as an intermediate domain, i.e. the normalized features are converted back to the spectral domain after normalization. It is implemented in MFCC frameworks. In [21], it is extended to operate on modulation spectra. Power functions used in PLP and PNCC could be seen as examples of \( q \)-log for certain \( q \)-values (\( q = 0.3 \) for PLP and \( q = 0.9 \) for PNCC). A previous study confirms that replacing power function with \( q \)-log could achieve similar performances [15]. In these works, a single \( q \), empirically chosen, is used. The use of multiple roots could benefit speech recognition because different speech units might have different sensitivity to distortions [22, 18]. Studies indicate that the use of multiple roots or an adaptive approach to determine \( q \) of \( q \)-MN have not yet be explored.

Some studies have found that mimicking human auditory systems in the feature extraction process benefits ASR [23, 24]. It is well known that humans do not have linear response to various frequency range. In general, humans are more sensitive to the changes of frequency of sounds in the low frequency regions than the changes in the high frequency regions. In feature extraction, we mimick human auditory systems by having more spectral components in the low frequency regions than in the high frequency regions. This can be done by mapping the spectral components into a non-linear scaled filter that represent human auditory systems. Some popular scales used are the Mel, Bark, and equivalent rectangular bandwidth (ERB) scales. MFCC uses the Mel scale while PLP implement the Bark scale. PNCC and gammatone cepstral coefficient (GFCC) use gammatone filter, which is based on the ERB scale. While it is evident that use non-linear
frequency scales benefit the speech recognition, the use of human-auditory inspired scales may not be optimum.

In this paper, we propose to use the \( q \)-log based feature normalization method on nonlinear scaled spectra. We use only Mel scale for this work. The objective is to see whether the use of non-linear frequency spacing could be optimized using \( q \)-log. We called it \( q \)-Mean Normalization (\( q \)-MN). We hope to address three issues here. The first is to investigate whether there could be any advantages of applying power functions, \( q \)-log in particular, on human-auditory inspired features on robustness of speech recognition. The second is to see the effect of implementing \( q \)-MN in the intermediate and non-intermediate domains, i.e. the normalized spectra are not converted back to the linear domain. The objective is to compare the effect of the various dynamics of linear, log, and power functions and their effect on robustness of the features in ASR. Thirdly, we propose an adaptive approach to to determine \( q \) for \( q \)-MN. The adaptive method is motivated by our analysis on the effect of applying \( q \)-MN on noisy speech. Our evaluation on the method finds that it is better than when applying single \( q \).

The remainder of this paper is organized as follows. In Sec. 2 we describe in detail the problem of noise robustness in speech recognition. In Sec. 3 \( q \)-log is briefly described and the effect of two normalization methods: mean normalization (MN) and \( q \)-MN on features are explained. We analyse the effect of \( q \)-MN on noisy speech in Sec. 4. Various front-ends used and the proposed method for adaptive \( q \)-MN are described in Sec. 5. The experimental setup and results are then discussed in Secs. 6 and 7 respectively. The paper is concluded in Sec. 8.

2. Problem Formulation

When ASR systems operate in real environments, they must deal with speech that is contaminated with noise. The different conditions of training and testing cause a mismatch and hence, degrades the performance of ASR. While adding noise information in the training can improve the performance of ASR, the resulting performances are still not satisfactory [9]. This is because, it is difficult to include all possible types and conditions of noise in the training.

The effect of noise on speech could be explained as follows. Let \( s(t) \) be the clean speech at time \( t \), which is corrupted by additive noise \( n(t) \) and convolutional noise \( h(t) \). Noisy speech \( y(t) \), i.e. speech corrupted by noise, is then given by the following in the time domain:

\[
y(t) = s(t) * h(t) + n(t).
\]

(1)

When extracting speech features, speech signals are first chunked with a fixed length, usually around 25-50 ms length. Each chunk, which is called a frame, is then windowed and transformed into the spectral domain by applying discrete Fourier transform (DFT). In speech recognition, speech features are usually derived from their power spectra, i.e. square of the magnitude. In this domain, the relation between noisy speech, clean speech and noise can be expressed as:

\[
P_Y(m, i) = P_S(m, i)P_H(m, i) + P_N(m, i) + 2M_S M_H M_N \cos \theta_{SN},
\]

(2)

where \( P_Y \), \( P_S \), \( P_H \), and \( P_N \) are the representation of \( y \), \( s \), \( h \), and \( n \) in the power spectral domain respectively and \( M_S \), \( M_H \), and \( M_N \) represent the magnitude spectra of \( s \), \( h \), and \( n \). The indexes \( m \) and \( i \) are frame and frequency indexes. The term \( 2M_S M_H M_N \cos \theta_{SN} \)
of Eq. (2) is called the cross-term. In many methods, it is ignored by assuming noise and speech are uncorrelated. By ignoring the cross term and denoting \( \lambda(m,i) = 1 + P_N(m,i)P_H(m,i) \), Eq. (2) can be written as:

\[
P_Y(m,i) = P_S(m,i)P_H(m,i)\lambda(m,i).
\]

(3)

To obtain the features, the power spectra are fed into human-auditory-inspired filterbanks such as Mel scale in MFCC and then log is applied on the output of the filterbanks. Eq. (3) can be represented in the log mel domain as:

\[
y(m,f) = s(m,f) + h(m,f) + \lambda(m,f),
\]

(4)

where \( y, s, h, \) and \( \lambda \) represent \( P_Y, P_S, P_H, \) and \( \lambda \) in the log mel domain and \( f \) is the index of the filterbank. The convolutional noise, \( h \), is fairly flat but \( \lambda \) is highly non-stationary, making it very difficult to remove.

3. Feature Normalization on \( Q \)-Log Based Features

\( Q \)-log is defined as:

\[
\log_q(x) = \frac{x^{1-q} - 1}{1-q}.
\]

(5)

Since \( \lim_{q \to 1} \log_q(x) = \log(x) \), \( q \)-log is a generalization of the natural logarithmic function. When \( q = 0 \), it is a linear function and when \( q = 1 \), it equals to the natural log function. The function lies between linear to log when \( 0 \leq q \leq 1 \). Its inverse, called \( q \)-exponential (\( q \)-exp), is defined as follows:

\[
e^q_x = (1 + (1 - q)x)^{1/(1-q)}.
\]

(6)

This function generalizes exponential function and when \( q = 1 \), it is the same as exponential functions.

\( Q \)-log is widely used in Tsallis statistics [25]. The function is used due to its non-additivity properties. \( Q \)-log does not have the same properties as log. It does not transform a multiplication operation into addition. A new set operators were proposed [26] to explain the properties of the \( q \)-log. They generalize multiplication, division, subtraction, and addition operators. Let \( +_q \) and \( -_q \) be the generalized addition and subtraction operators. They are defined as:

\[
a +_q b = a + b + (1-q)ab,
\]

(7)

and

\[
a -_q b = \frac{a - b}{1+(1-q)b}.
\]

(8)

These operators are the same as the standard operators when \( q = 1 \). Based on these operators, some properties of \( q \)-log can be written as follows:

\[
\log_q(ab) = \log_q(a) +_q \log_q(b),
\]

(9)
and
\[
\log_q \left( \frac{a}{b} \right) = \log_q(a) - q \log_q(b). \tag{10}
\]

These properties make it clear that $q$-log is non-additive when $q \neq 1$. For more details about these operators, please refer to \[25\].

In Tsallis statistics, the non-additivity of $q$-log and $q$-exp is used to explain non-extensive phenomena in many complex systems \[25\] \[20\]. Under this framework, an entropy is defined \[25\] and various distributions are derived such as $q$-Gaussian and $q$-exponential distributions \[27\] \[28\]. With this framework, many non-extensive phenomena of many complex systems in physics, biology, economy, finance, etc, can be explained. In this framework, the parameter, $q$, is usually chosen empirically. Unfortunately, a method to choose $q$ has not yet been defined in the proposed studies and most studies only select $q$ that best fit the phenomena of interest.

Tsallis statistical frameworks have also been used in speech recognition and shown improved recognition accuracies \[29\] \[15\] \[30\]. One implementation of Tsallis statistics is the use of $q$-log instead of the log function in the feature extraction process \[19\] \[15\]. When $q$-log is applied on noisy speech, Eq. (2) can be expressed in $q$-log domain as the following (for readability, indexes $m$ and $k$ are dropped):
\[
y_q = s_q + h_q + \lambda_q, \tag{11}
\]
where $y_q$, $s_q$, $h_q$, and $\lambda_q$ are the $q$-log of $P_Y$, $P_S$, $P_H$, and $\lambda$ respectively. We can expand (11) as:
\[
y_q = s_q + h_q + \lambda_q + (1 - q)s_qh_q + (1 - q)s_q\lambda_q + (1 - q)h_q\lambda_q + (1 - q)^2 s_qh_q\lambda_q. \tag{12}
\]
When $q \neq 1$, it is obvious that $s_q$, $h_q$, and $\lambda_q$ are non-additive. When $s_q > 0$, the third, fourth, and fifth terms of Eq. (12) would likely be positive and using $q$-log could actually increase the mismatch when features are not normalized, which has been confirmed in a previous study \[31\].

Mean normalization (MN) is a simple and powerful technique to improve robustness in speech recognition systems. It is effective to remove stationary distortions such as convolutional noise and white noise. Cepstral mean normalization (CMN) and log spectral mean normalization (LSMN) are two examples of MN methods. In MN, the objective is to make the long term mean of the features to be zero. Since convolutional noise is relatively stationary, subtracting the long-term average from the features is effective to remove it. But, MN has limited effectiveness on removing $\lambda$.

When MN is applied in the $q$-log domain, assuming that speech and noise are uncorrelated, $h_q$ is stationary, and $\lambda$ has zero mean (Need to be noted that the assumption of $\overline{\lambda} = 0$ may not be true. But, our empirical observation shows that it is very close to zero), i.e. $\overline{\lambda} = 0$, we obtain the mean normalized features of noisy speech, $\tilde{y}_q$, as follows:
\[
\tilde{y}_q = y_q - \overline{y}_q = \left(1 + (1 - q)c_q\right)(\bar{s}_q + q\bar{\lambda}_q)
= \left(1 + (1 - q)c_q\right)\left[(1 + (1 - q)\bar{\lambda}_q)s_q + \lambda_q\right]. \tag{13}
\]
where $c_q = \mathbb{E}\{h_q\}$. As indicated in Eq. (13), there are two things that could be observed. First, the term $(1 + (1 - q)c_q)$ could amplify the features, which might increase the mismatch and hamper the performance of ASR even when we assume the convolutional noise to be stationary. Usually, the power spectra are normalized before applying
MN as in PNCC [6]. Second, the term of \((1 + (1 - q)\lambda_q)\) might benefit ASR since it amplifies speech and as the results, noise could be masked, reducing the effect of noise. Overall, since both terms are multiplicative to speech, the performance of MN could be very sensitive to the choice of \(q\) and big changes in performance may occur for small changes of \(q\) [31].

A more appropriate way to normalize \(q\)-log based features is proposed [19]. The method is called \(q\)-LSMN. In that work, the method was applied in on the power spectra. It was also performed as an intermediate processing step, i.e. the normalized spectra were transformed back to the spectral domain after normalization. In this paper, we apply the method on the output of human-auditory-inspired filterbanks. We call it \(q\)-mean normalization (\(q\)-MN). It is formulated as:

\[
\hat{y}_q = \frac{y_q - \overline{y}_q}{1 + (1 - q)\overline{y}_q} = y_q - q\overline{y}_q \tag{14}
\]

Applying \(q\)-MN on noisy speech, using the same assumptions as before, we obtain the normalized features, \(\hat{y}_q\), as:

\[
\hat{y}_q = \frac{\hat{s}_q + \lambda_q + (1 - q)\hat{s}_q \lambda_q}{1 + (1 - q)\overline{s}_q} = s_q + \frac{1 + (1 - q)\hat{s}_q}{1 + (1 - q)\overline{s}_q} \lambda_q, \tag{15}
\]

where \(s_q = \sum_{m=1}^{M}s_q(m)\) is the arithmetical mean of \(s_q\) and \(\overline{s}_q\) is the mean normalized version of \(s_q\), i.e. \(\overline{s}_q = s_q - \overline{y}_q\), and \(\hat{s}_q\) is the normalized features of clean speech after \(q\)-MN. Based on Eq. (15), \(q\)-MN is the same as MN when \(q = 1\) or when \(s_q\) has zero mean. Interestingly, assuming that the convolutional noise is stationary, \(c_q\) is removed. Therefore, power normalization as in PNCC is not needed. In addition, Eq. (15) shows that \(\lambda_q\) is affected by the factor \(\frac{1 + (1 - q)\overline{s}_q}{1 + (1 - q)\overline{y}_q}\). For unvoiced consonants in speech, \(\lambda_q\) is oppressed since generally \(\hat{s}_q < \overline{s}_q\) for consonants. But, this also means that \(\lambda_q\) is amplified when \(\hat{s}_q > \overline{s}_q\), as is the case for vowels and voiced consonants. Fortunately, since \(\hat{s}_q\) is the mean subtracted features, we could assume that most of the normalized spectra have lower energy than the average spectra before normalization, making \(\lambda_q\) is suppressed when \(q < 1\) is used.

4. The Effect of Q-LSMN on Noisy Speech

In this section, we present our analysis on preliminary results of the effect of \(q\)-MN on noisy speech. We observe its effect on the log mel spectral distance (LSD).

We investigated the effect of \(q\)-MN on noisy speech by applying it on 250 utterances of test set A of Aurora-2. In Aurora-2, each utterance is corrupted by 4 types of noise (Subway, Babble, Car, and Exhibition) artificially for SNR conditions of -5 dB to 20 dB. So, there are total 1000 utterances for each SNR condition. For each utterance, we first subtracted the spectra with their mean so we could safely assume that the effect of constant distortion, which could be caused by channel distortion and stationary noise was removed. Then, \(q\)-MN was performed using fixed \(q\) varied from \(0 \leq q \leq 1\) on the normalized spectra. \(q\)-MN is applied in mel domain. We performed the same process on both clean and noisy speech. LSD between clean and noisy speech was calculated and then averaged over all frames for all utterances.

Figure 1 shows the LSD after \(q\)-MN. The results are averaged over all SNR conditions (-5 to 20 dB). We found that \(q\)-MN achieved the lowest LSD when \(q = 1\) (See
5. Proposed Method

In this work, we would like to investigate three issues. First is to investigate the benefit of $q$-log function on human-auditory inspired features on robustness of speech recognition. So, we will compare the robustness of features when applying $q$-MN after human-auditory inspired scales. It is known that humans have nonlinear response to sounds. They are more sensitive to the changes of frequency in the low frequency than in the high frequency. Some studies have shown that mimicking human auditory systems in the feature extraction process improves the robustness of ASR [32]. One of most commonly used human auditory scales is the Mel scale.

Mel is originated from the word melody. The Mel scale, which is introduced in [33], is a scale of pitch that is determined by listeners perceptually to have an equal distance from each other. It is used in MFCC, which is arguably the most common features for ASR. The Mel scale can be approximated with the following formula:

$$\text{Mel}(f) = 2595 \log_{10} \left( 1 + \frac{f}{700} \right)$$  \hspace{1cm} (16)$$

Where $f$ is the frequency in Hz and Mel is its corresponding value in Mel scale.

Secondly, we would like to see the effect of implementing $q$-MN in the intermediate i.e. the features were transformed back to the filterbank domain before applying DCT, and the non-intermediate domains, i.e. then normalized features were fed directly to discrete cosine transform (DCT). The objective is to compare the effect of the various dynamics of linear, log, and power functions and their effect on robustness of the features in ASR.

Figure 2 shows the block diagrams of the front-ends that we evaluate to address both aforementioned issues. There are total of four front-ends. They consist of linear
and mel filterbanks and for each filterbank, $q$-MN is applied in the intermediate and non-intermediate domains.

Thirdly, we would like to proof that using a different $q$ for different parts of speech is effective to improve the robustness of speech recognition. As shown in previous section, applying a different $q$ for different parts of speech is effective in reducing the LSD, and hence it might also benefit to the performance of speech recognition. These results motivate us to develop an adaptive technique to determine a suitable choice for $q$. In this paper, we use only two values of $q$ which will be applied for spectral valleys and spectral peaks. We denote $q_p$ for $q$ to be applied for the spectral peaks while $q_v$ is the values of $q$ for the spectral valleys. The decision whether particular spectra are the peaks or the valleys are determined by whether the spectra are higher or lower than the long-term means, which are pre-calculated offline. Since the reduction of LSD may not necessarily correspond to the accuracy improvements, we evaluate several combinations of $q_p$ and $q_v$. We varied $0 \leq q_p \leq 0.7$ and $0.5 \leq q_v \leq 1$. After a mel spectrum is decided to be a peak or a valley, the appropriate $q$ value is selected. Then, the speech spectra are transformed into the $q$-log mel spectral domain and $q$-MN is applied accordingly. The normalized features are then transformed back into the mel spectral domain. We denote the resulting features

![Fig. 2. The block diagrams of the evaluated front-ends.](image-url)
Fig. 3. Block diagram of the proposed front-end.

As $q$-MFCC$_A$. Figure 3 shows the proposed feature extraction chain for $q$-MFCC$_A$.

For all features, a “standard” signal processing method is applied to the speech signals to obtain their power spectra. First, the signals are passed through a pre-emphasis filter, then they are windowed using a Hamming window with 25 ms length and 10 ms frame shift. Then, discrete Fourier transform (DFT) is applied to each frame to transform the signals into the frequency domain. The magnitude of the spectra is squared to obtain the power spectra. The power spectra are then fed into the filterbanks. The linear filterbank accumulate 256 frequency components of power spectra into 23 uniformly divided filterbanks. For Mel filterbanks, the 256 frequency components are also accumulated into 23 filterbanks. But instead of being uniformly distributed according to a linear scale in the frequency domain, the filters are distributed according to the Mel scale. In other words, more filterbank components are put in the low frequency regions than in the high frequency regions.

No noise removal technique was applied so the improvements that were achieved were the results of applying of $q$-log transformation, the effect of human auditory scales and the normalization. For references, results for several standard features (LFCC, MFCC) with cepstral mean normalization (CMN) were provided.
6. Experimental Setup

All features were evaluated using Aurora-2 database \[34\]. In this speech database, eight types of additive noise and two types of convolutional noise are used to artificially create the noisy data. There are three test sets and noise is added artificially at various SNR: 20 dB, 15 dB, 5 dB, 0 dB and -5 dB. In this paper, we used only the clean data to train the acoustic models for ASR.

For the speech recognition, the standard HMM-based ASR system provided in the corpus was used. Each digit was modeled by 16 states HMM, left-to-right where each state was modelled using GMM with three Gaussian components. Two pause models: sil and sp were used: “sil” and “sp” models. The “sil” model had 3 states with 6 Gaussian components while “sp” had a state tied to the middle state of the sil model. The features had 39 dimensions: 13 static features including the zeroth cepstral, 13 first derivatives, and 13 second derivatives.

For evaluation metrics we used word accuracy (WA). It is computed as follows:

\[
WA = \frac{H - I}{N},
\]

where \(H\) is the number of correctly recognized words, \(I\) is the number of insertions, and \(N\) is the number of words. For Aurora-2, it is common to use average word accuracy as a metric. It is computed by averaging the word accuracy for SNR conditions 0 to 20 dB (clean and -5 dB are excluded).

7. Results and Discussions

In this section, several evaluations on the investigated features are given. First, the effect of using different human auditory models is shown and analysed. Secondly, the effect of applying \(q\)-log in intermediate and non-intermediate domains is evaluated. Thirdly, \(q\)-MFCC is evaluated.

7.1 The effect of linear vs nonlinear scales

Figure 4 compare the performance of \(q\)-MN when it is applied in linear and Mel scales. As expected, applying \(q\)-log in the linear domain has the worse performance compared to Mel scale. This confirms the importance of using human-auditory information for extraction of speech features. The performances of \(q\)-MFCC are consistently better than \(q\)-LFCC in both intermediate and non-intermediate domains.

7.2 Intermediate vs non-intermediate domains

As displayed in Fig. 5 the best \(q\)-values are relatively similar for all SNR conditions when \(q\)-MN is applied in the intermediate domain but it is evident that the optimal value of \(q\) is sensitive to the SNR conditions when \(q\)-MN is applied in the non-intermediate domain (See Fig. 6). The best \(q\) is closer zero for low SNR. But, the performance in high SNR conditions is degraded compared to when \(q = 1\). As can be seen in Eq. (15), applying \(q\)-MN causes noise to be affected by the ratio between speech with its long term average when \(q < 1\). In the low SNR conditions, speech is dominated by noise in low SNR. As the consequences, the ratio is closer to zero when \(q\) is small, and the noise spectra are suppressed. On the other hand, speech is more dominant than noise in the high SNR conditions. Therefore, applying low \(q\) causes the noise spectra to be amplified. This might explain the degraded of performance in high SNR conditions.
Vowels and voiced consonants lie at low frequency regions and have high energy. Therefore, the compression of the dynamics of the features in these regions influence
the robustness of the features. Recall that when \( q \) is close to zero, the \( q \)-log function approaches the linear function. This would be similar to the use of linear filterbank, which has been known to be less effective. On the other hand, unvoiced consonants usually lie in the high frequency region and have low energy. When a high \( q \) is used, the dynamics of the transformed spectra is larger in low energy regions, making it more sensitive to noise. Therefore, using \( q \) close to 1 could cause large drops in the performance for high SNR conditions. Meanwhile, for speech parts that lie in high energy regions such as vowels, it is very likely that their energy is higher than the average spectra. As the results, they mask the noise parts and hence, these regions are more robust to noise. So, increasing the dynamics would have less effect.

We observe that when \( q \)-log is applied in the intermediate domain, the best performances are achieved at quite similar values of \( q \) for all SNR conditions. This is because, when the normalized spectra is converted back to the Mel domain, the ratio between speech and noise will be the same as before normalization. Hence, the \( q \)-values may have the same effect on all SNR conditions. The improvements might be due to the non-additive properties of \( q \)-log.
Fig. 7. The average word accuracy (%) for $q$-MFCC$_A$ for various combinations of $q_p$ and $q_v$. The average is calculated over 0 to 20 dB SNR. $q$-MFCC$_A$ is compared when single $q$ is used ($q$-MFCC$_I$). The performance of $q$-MFCC$_I$ for various $q$ is shown in solid line.

Table 1. The performance comparison of $q$-log based features: $q$-LFCC and $q$-MFCC, and reference features: LFCC and MFCC. The reference features are normalized using CMN.

<table>
<thead>
<tr>
<th>Features</th>
<th>Clean</th>
<th>20</th>
<th>15</th>
<th>10</th>
<th>5</th>
<th>0</th>
<th>-5</th>
<th>Ave</th>
</tr>
</thead>
<tbody>
<tr>
<td>LFCC+CMN</td>
<td>99.05</td>
<td>96.73</td>
<td>91.99</td>
<td>77.57</td>
<td>47.98</td>
<td>23.93</td>
<td>13.14</td>
<td>67.64</td>
</tr>
<tr>
<td>$q$-LFCC$_I$ ($q = 0.9$)</td>
<td><strong>99.09</strong></td>
<td><strong>96.73</strong></td>
<td><strong>92.95</strong></td>
<td><strong>81.90</strong></td>
<td><strong>59.07</strong></td>
<td><strong>30.79</strong></td>
<td><strong>14.05</strong></td>
<td><strong>72.29</strong></td>
</tr>
<tr>
<td>$q$-LFCC$_N$ ($q = 0.4$)</td>
<td>91.54</td>
<td>90.66</td>
<td>87.50</td>
<td>80.19</td>
<td>63.40</td>
<td><strong>36.55</strong></td>
<td><strong>14.75</strong></td>
<td>71.66</td>
</tr>
<tr>
<td>MFCC+CMN</td>
<td>99.07</td>
<td>97.04</td>
<td>92.90</td>
<td>79.47</td>
<td>49.22</td>
<td>24.16</td>
<td>13.44</td>
<td>68.56</td>
</tr>
<tr>
<td>$q$-MFCC$_I$ ($q = 0.8$)</td>
<td><strong>99.10</strong></td>
<td><strong>97.32</strong></td>
<td><strong>94.45</strong></td>
<td><strong>86.50</strong></td>
<td><strong>67.46</strong></td>
<td><strong>39.19</strong></td>
<td><strong>17.15</strong></td>
<td><strong>76.99</strong></td>
</tr>
<tr>
<td>$q$-MFCC$_N$ ($q = 0.3$)</td>
<td>91.90</td>
<td>91.64</td>
<td>89.59</td>
<td>84.55</td>
<td><strong>70.51</strong></td>
<td><strong>43.08</strong></td>
<td>15.26</td>
<td>75.87</td>
</tr>
<tr>
<td>$q$-MFCC$_A$</td>
<td><strong>99.10</strong></td>
<td><strong>97.70</strong></td>
<td><strong>95.74</strong></td>
<td><strong>90.75</strong></td>
<td><strong>75.76</strong></td>
<td><strong>44.70</strong></td>
<td><strong>18.33</strong></td>
<td><strong>80.93</strong></td>
</tr>
</tbody>
</table>

7.3 The performance of $q$-MFCC$_A$

Figure [7] shows the performance of ASR for $q$-MFCC$_A$ for various combination of $q_p$ and $q_v$. Reasonably good results were obtained when $0.4 \leq q_p \leq 0.7$ and $0.7 \leq q_v \leq 1$.

It is obvious that $q$-MFCC$_A$ achieves a better performance than using single $q$ ($q$-MFCC$_I$). For comparison, performance of $q$-MFCC$_I$ is displayed in solid line. The highest accuracy is achieved when $q_p = 0.6$ and $q_v = 0.9$. This set of values are different with the LSD presented previously. This not so surprising since the relation between the reduction on spectral distortion and the improvement on ASR performance is not always linear.

The performance of the $q$-log based features are compared to several references fea-
tures: MFCC and LFCC in Table I. All the reference features are normalized using CMN. It is evident that $q$-MN is more effective than CMN since the $q$-log features achieved better performance for both linear and Mel scales. On average, $q$-MFCC$_I$ achieves 12.3% relative improvement over MFCC and $q$-LFCC$_I$ achieves 6.87% improvement over LFCC. As shown previously, $q$-MFCC$_I$ and $q$-LFCC$_I$ are better at high SNR conditions but $q$-MFCC$_N$ and $q$-LFCC$_N$ are better at low SNR.

$Q$-MFCC$_A$ ($q_p = 0.6, q_v = 0.9$) has the best accuracy over all evaluated features. An improvement of 19.65% and 5.12% are achieved over MFCC and $q$-MFCC$_I$ (the case of using single $q$) respectively. $Q$-MFCC$_A$ consistently better for all SNR even when we compare it with $q$-MFCC$_N$ for low SNR conditions. As we have explained previously, the results indicate that $q$-MN is able to suppress noise when low values of $q$ are used. In low SNR conditions, noise is dominant. Noise parts may dominate the peaks of the noisy speech spectra, and hence, applying a $q$ closer to zero would generally reduce their energy. But, if we apply a value of $q$ closer to zero for spectral valleys as well, the accuracy would suffer since noise would be amplified. So, when a value of $q$ closer to 1 is applied for these regions, we could maintain to suppression of noise in spectral valleys. This is important especially when the SNR is low. Therefore, the proposed method is able to improve the performance of speech recognition in both low and high SNR conditions.

8. Conclusion

In this paper, we have shown that applying $q$-MN to human-auditory filterbanks improves the performance of speech recognition in noisy environments. The experimental results suggest that $q$-MN is more effective when it is applied in an intermediate domain for high SNR conditions. But, for low SNR conditions, the results indicate that more robust features can be obtained when the normalized features are not transformed back from this domain but used as they are in the remaining features extraction steps. When the SNR is low, the ratio between noise and speech could be minimized using low $q$ in the $q$-log domain. This is not the case when they are inverted back to the linear domain. Compared to the standard features such as MFCC, $q$-log based features are largely better.

We also proposed an adaptive approach to determine $q$ of $q$-MN. We found that $q$-MN reduced the spectral distortion on the peaks of spectra when $q < 1$ was applied. However, it also increased the distortion in spectral valleys. Hence, we applied two $q$ values for $q$-MN and found that $q_p = 0.6$ and $q_v = 0.9$ achieves the best accuracy. Need to be noted that there is no conclusive proof that these values work best for all noisy conditions and further studies are required to find the correlation between physical properties of speech with $q$. Nonetheless, Our experimental results demonstrated the effectiveness of multiple $q$-values compared to a single fixed one.

REFERENCES

3. S. Abd El-Moneim, M. I. Dessouky, F. E. Abd El-Samie, M. A. Nassar, and M. Abd El-Naby, “Hybrid speech enhancement with empirical mode decomposition and spec-


