IMPROVED CEPSTRAL MEAN AND VARIANCE NORMALIZATION USING BAYESIAN FRAMEWORK

N. Vishnu Prasad, S. Umesh

Department of Electrical Engineering, Indian Institute of Technology - Madras, India

ABSTRACT

Cepstral Mean and Variance Normalization (CMVN) is a computationally efficient normalization technique for noise robust speech recognition. The performance of CMVN is known to degrade for short utterances, due to insufficient data for parameter estimation and loss of discriminable information as all utterances are forced to have zero mean and unit variance. In this work, we propose to use posterior estimates of mean and variance in CMVN, instead of the maximum likelihood estimates. This Bayesian approach, in addition to providing a robust estimate of parameters, is also shown to preserve discriminable information without increase in computational cost, making it particularly relevant for Interactive Voice Response (IVR)-based applications. The relative WER reduction of this approach w.r.t. Cepstral Mean Normalization, CMVN and Histogram Equalization are (i) 40.1%, 27% and 4.3% with the Aurora2 database for all utterances, (ii) 25.7%, 38.6% and 30.4% with the Aurora2 database for short utterances, and (iii) 18.7%, 12.6% and 2.5% with the Aurora4 database.

Index Terms— Robust speech recognition, CMVN, HEQ, VTS, Bayesian estimation.

1. INTRODUCTION

The performance of speech recognition systems degrade under noisy environments due to mismatch between training and test conditions. Numerous techniques have been published in the literature to address this issue [1, 2, 3, 4, 5, 6, 7]. These techniques can be broadly classified into two categories; *Model Adaptation* and *Feature Normalization*. Model adaptation techniques adapt the trained models to match the test utterance, whereas feature normalization techniques modify the noisy test features to match the statistics of the clean training features.

Feature normalization techniques can be further categorized as parametric and non-parametric approaches. In this paper we focus on parametric feature normalization techniques; specifically, Cepstral Mean Normalization (CMN) [1], Cepstral Mean and Variance Normalization (CMVN) [3, 8] and Quantile-based Histogram Equalization (HEQ) [4, 5].

CMN was initially proposed to compensate the channel effects in the form of convolutive noise [1]. CMN matches the first order moment of every training and test utterance by removing their respective time average, and transforming each utterance to zero mean.

CMVN matches both mean and variance by transforming every training and test utterance to zero mean and unit variance. This paper focuses on conventional per-utterance CMVN and not the segmental version of CMVN [3, 8, 9], although the methods proposed in this paper can be easily extended to segmental-CMVN. Let $\mathbf{x}_t$ denote a 13-dimensional cepstral vector at time t of an utterance, and $x_t(i)$ represent the i^{th} component of $\mathbf{x}_t$. Let $X = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_t, \cdots, \mathbf{x}_T]$ denote an utterance of length T. CMVN is performed by first computing the mean (μ) and variance (σ^2) independently for every dimension in maximum likelihood (ML) framework, as shown below.

$$\mu_{ML}(i) = \frac{1}{T}\sum_{t=1}^{T} x_t(i) \quad 1 \le i \le 13 \tag{1}$$

$$\sigma^2_{ML}(i) = \frac{1}{T-1}\sum_{t=1}^{T} (x_t(i) - \mu_{ML}(i))^2 \quad 1 \le i \le 13 \tag{2}$$

The mean and variance normalized frame $\hat{\mathbf{x}}_t$ is computed for all 13 dimensions and for all frames as shown in Eq. (3) to obtain the normalized utterance $\hat{X}$.

$$\hat{x}_t(i) = \frac{x_t(i) - \mu_{ML}(i)}{\sigma_{ML}(i)} \quad 1 \le t \le T,\ 1 \le i \le 13 \tag{3}$$

HEQ can be considered as an extension to CMN and CMVN. HEQ equalizes the training and test utterances to match the statistics of a reference cumulative distribution function (cdf), thus matching the higher order moments [4, 5].

1.1. Motivation

CMN, CMVN and HEQ techniques work best for long utterances. The performance of HEQ is better compared to CMVN and CMN, as HEQ matches all the moments of the training and test utterance; whereas CMVN matches only the mean and variance, and CMN matches only the mean.

The performances of CMVN and HEQ degrade for short utterances due to lack of sufficient data for parameter estimation [10]. While CMVN estimates only mean and variance, HEQ estimates the entire cdf of the utterance before normalization. The estimate of these parameters are therefore not robust for short utterances leading to performance degradation.

Further, there is also loss of discriminable information when CMVN and HEQ are applied for short utterances [10]. CMVN forces all utterances to transform to zero mean and unit variance; i.e., the mean and the variance of every dimension of an utterance after CMVN are zero and one respectively. In the case of HEQ, all the utterances are forced to match the reference cdf. The mean and the variance of an utterance after HEQ would be the reference mean and the reference variance respectively. Since every utterance and all its feature components are forced to have the same statistics, some amount of discriminable information is lost in the process of normalization.

Other algorithms like the vector Taylor series (VTS) work for short utterances [7], but are computationally expensive [11]. Hence an improvement over simple approaches such as CMN, CMVN and HEQ could still be important for real-time Interactive Voice Response (IVR) applications.

1.2. Proposed Method

As discussed above, there is not enough data to obtain robust estimate of parameters for short utterances. We propose to use a Bayesian framework for parameter estimation to address this problem. A Bayesian framework compensates for insufficient data by having a prior distribution for the parameters to be estimated. This prior knowledge is then used during parameter estimation to compute the posterior estimates.

In this work, we use the Bayesian framework in the conventional per-utterance CMVN. We propose to use a posterior estimate of mean and variance ($\mu_{post}(i)$ and $\sigma^2_{post}(i)$) in CMVN instead of the maximum likelihood estimates. The choice of the prior distribution and the method to estimate the posteriors are explained in section 2. These posterior parameters are then used to perform mean and variance normalization to obtain the normalized utterance $\check{X}$ as shown below.

$$\check{x}_t(i) = \frac{x_t(i) - \mu_{post}(i)}{\sigma_{post}(i)} \quad 1 \leq t \leq T,\ 1 \leq i \leq 13 \qquad (4)$$

We use the term *Bayesian-CMVN* (BCMVN) to denote this method of using Bayesian estimates for performing CMVN.

Our analysis (section 3) shows that the proposed method, in addition to computing robust estimates, is also able to retain some amount of utterance-specific and dimension-specific information compared to CMVN and HEQ. This improves the recognition performance of BCMVN over CMN, CMVN and HEQ, and the results of the experiments (section 6) indicate that this improvement is not just for short utterances, but also for long utterances. In the Aurora2 database, the relative word error rate (WER) reduction of BCMVN w.r.t. CMN, CMVN and HEQ for all utterances are 40.1%, 27% and 4.3% respectively, and 25.7%, 38.6% and 30.4% respectively for short utterances. The WER of BCMVN with the Aurora4 database is also consistently lesser compared to CMN, CMVN and HEQ, and the relative reduction is 18.7%, 12.6% and 2.5% respectively.

The rest of the paper is organized as follows. Section 2 explains the methodology of choosing the prior distribution and the BCMVN algorithm. This is followed by detailed analysis in section 3, on how BCMVN is able to preserve discriminable information compared to CMVN and HEQ. In section 4, we present a modified version of the BCMVN. The experimental setup is explained in section 5 and the results are discussed in section 6. Finally, conclusions and future work are presented in section 7.

2. BAYESIAN CEPSTRAL MEAN AND VARIANCE NORMALIZATION (BCMVN)

In this section, we discuss the implementation of the proposed BCMVN algorithm. The parameters that are of interest for performing CMVN are $\mu(i)$ and $\sigma^2(i)$ for all dimensions $1 \leq i \leq 13$. Since the parameters for each dimension are estimated independently and in the same manner, all further discussion is made for one particular feature component. The dimension index i is dropped hereafter for notational convenience. It is to be noted that non-bold symbols indicate one feature component of the corresponding vector counterpart.

The parameters that are estimated for performing conventional CMVN are μ and σ^2. The maximum likelihood estimate of these parameters are obtained by maximizing the likelihood of the parameters (μ, σ^2) w.r.t. the data (X); i.e.,

$$(\mu_{ML}, \sigma^2_{ML}) = \underset{\mu,\sigma^2}{argmax}\ \ p(X; \mu, \sigma^2)$$

To estimate these parameters in an ML framework, each dimension of the input cepstral feature is assumed to be Gaussian distributed [9]. Then, the ML estimates of mean and variance for a Gaussian distribution can be computed using Eq. (1) and Eq. (2).

A Bayesian approach is used to obtain robust estimates of parameters when data are insufficient or are corrupted by noise. Bayesian estimation treats the parameters to be estimated as random variables with a prior probability density function (pdf) $p(\mu, \sigma^2)$. One method to estimate the parameters using the Bayesian framework is to choose the *mean of the posterior distribution* as the estimate of the parameters. This estimate is also known as the minimum-mean-square-error (MMSE) estimate. The posterior distribution of the parameters is given by

$$p(\mu, \sigma^2|X) = \frac{p(X|\mu, \sigma^2)\, p(\mu, \sigma^2)}{p(X)}$$

and the MMSE estimate of parameters are

$$(\mu_{post}, \sigma^2_{post}) = \mathbb{E}\left[\mu, \sigma^2|X\right]$$

where $\mathbb{E}$ is the expectation operator. The method of choosing the prior distribution is discussed next.

2.1. Choosing the Prior Distribution

The choice of the prior distribution is based on two factors, (i) domain knowledge and (ii) mathematical tractability. As in the case of CMVN, we assume that each component of the cepstral features follows a Gaussian distribution. Now, a prior distribution has to be chosen for the parameters of the Gaussian random variable. Many kinds of priors for the parameters of a Gaussian distribution are studied in the literature and a compilation is available [12].

In this work, we choose a joint conjugate prior distribution $p(\mu, \lambda)$ for the mean μ and the precision $\lambda = \frac{1}{\sigma^2}$ for mathematical convenience. Normal-Gamma conjugate prior with four parameters $(\mu_0, \kappa_0, \alpha_0, \beta_0)$ is chosen.

$$NG(\mu, \lambda; \mu_0, \kappa_0, \alpha_0, \beta_0) \sim \mathcal{N}(\mu; \mu_0, (\kappa_0\lambda)^{-1})\Gamma(\lambda; \alpha_0, \beta_0) \quad (5)$$

where $\mathcal{N}$ represents a Gaussian pdf as the prior for the mean μ given λ with parameters (μ_0, κ_0), and Γ represents a Gamma pdf for the precision λ with parameters $(\alpha_0, rate = \beta_0)$ [12].

The parameters $\mu_0, \kappa_0, \alpha_0, \beta_0$ are estimated using the training data in the maximum likelihood framework by maximizing the log likelihood of Eq. (5). Let N denote the total number of training utterances. Let μ_n and λ_n denote the mean and precision of the n^{th} training utterance. Now, μ_0 and κ_0 can be derived as follows.

$$\mu_0 = \frac{\sum_{n=1}^{N} \mu_n\lambda_n}{\sum_{n=1}^{N} \lambda_n} \qquad (6)$$

$$\kappa_0 = \frac{N}{\sum_{n=1}^{N} \lambda_n(\mu_n - \mu_0)^2} \qquad (7)$$

The parameters α_0 and β_0 can be estimated by fitting a Gamma pdf to the training precision values, i.e., $\{\lambda_n\}\ 1 \leq n \leq N$. We use MATLAB command *gamfit* to estimate these parameters. It is to be noted that, MATLAB estimates the scale parameter of the Gamma distribution instead of the rate β_0. The rate parameter β_0 is obtained by inverting the scale parameter [12]. The prior parameters $\mu_0, \kappa_0, \alpha_0, \beta_0$ are estimated for all the 13 cepstral feature dimensions independently.

Table 1: CMVN vs BCMVN

Notation: $X = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_t, \ldots, \mathbf{x}_T]$ - an utterance of length T; $\mathbf{x}_t$ - 13 dimensional cepstral vector at time instance t; $x_t(i)$ - i^{th} component of $\mathbf{x}_t$

CMVN	**BCMVN**
Normalized frame $\hat{\mathbf{x}}_t$ is obtained as	Normalized frame $\check{\mathbf{x}}_t$ is obtained as
$\hat{x}_t(i) = \frac{x_t(i) - \mu_{ML}(i)}{\sigma_{ML}(i)} \quad 1 \le t \le T,\ 1 \le i \le 13$ (8)	$\check{x}_t(i) = \frac{x_t(i) - \mu_{post}(i)}{\sigma_{post}(i)} \quad 1 \le t \le T,\ 1 \le i \le 13$ (11)
where	where
$\mu_{ML}(i) = \frac{1}{T}\sum_{t=1}^{T} x_t(i) \quad 1 \le i \le 13$ (9)	$\mu_{post}(i) = \frac{\kappa_0(i)\mu_0(i) + T\mu_{ML}(i)}{\kappa_0(i) + T}$ (12)
$\sigma^2_{ML}(i) = \frac{1}{T-1}\sum_{t=1}^{T}(x_t(i) - \mu_{ML}(i))^2 \quad 1 \le i \le 13$ (10)	$\sigma^2_{post}(i) = \frac{\beta_0(i) + \frac{T}{2}\sigma^2_{ML}(i) + \frac{\kappa_0(i)T(\mu_{ML}(i) - \mu_0(i))^2}{2(\kappa_0(i)+T)}}{\alpha_0(i) + \frac{T}{2}}$ (13)
	$\mu_0(i), \kappa_0(i), \alpha_0(i), \beta_0(i)$ are estimated from the training data

2.2. BCMVN Algorithm

As in the case of CMVN, BCMVN is also applied for both training and test utterances to reduce the mismatch between the two. Given an utterance X (training or test) of length T, the ML estimate of the mean (μ_{ML}) and the variance (σ^2_{ML}) are first computed using Eq. (9) and Eq. (10) respectively, as shown in Table 1.

Now, the posterior pdf $p(\mu, \lambda|X)$ is also of the same form as the prior pdf, but with different parameters $\mu_p, \kappa_p, \alpha_p, \beta_p$ [12]. The posterior pdf is given by

$$p(\mu, \lambda|X) \sim NG(\mu, \lambda|\mu_p, \kappa_p, \alpha_p, \beta_p) \tag{14}$$

The parameters $\mu_p, \kappa_p, \alpha_p, \beta_p$ are estimated using the ML estimates (μ_{ML}, σ^2_{ML}), the utterance length T and the prior parameters ($\mu_0, \kappa_0, \alpha_0, \beta_0$) as shown next.

$$\mu_p = \frac{\kappa_0\mu_0 + T\mu_{ML}}{\kappa_0 + T} \tag{15}$$

$$\kappa_p = \kappa_0 + T, \quad \alpha_p = \alpha_0 + \frac{T}{2} \tag{16}$$

$$\beta_p = \beta_0 + \frac{T}{2}\sigma^2_{ML} + \frac{\kappa_0 T\,(\mu_{ML} - \mu_0)^2}{2(\kappa_0 + T)} \tag{17}$$

These parameters are estimated for all the 13 dimensions using the respective priors and the ML estimates.

Then, the posterior estimates or MMSE estimates of the mean and the variance are obtained as the mean of the marginal pdfs of μ and λ. The marginal distribution of mean μ is a Student's *t*-distribution, and for precision λ it is a Gamma distribution [12]. The MMSE estimate of the mean and the variance is given by

$$\mu_{post} = \mu_p \quad \sigma^2_{post} = \frac{1}{\lambda_{post}} = \frac{\beta_p}{\alpha_p} \tag{18}$$

and the expressions are shown in column 2 of Table 1 (Eq. (12) and Eq. (13)). These posterior estimates are then used to perform the BCMVN as shown in Eq. (11) in Table 1.

It is to be noted that the BCMVN algorithm is different from the other variations of CMVN proposed to estimate the parameters [3, 8, 9]. These methods implement segmental CMVN and use the estimate of parameters from the previous utterances or frames to obtain the current estimate. BCMVN is proposed as a per-utterance normalization where the entire utterance is available. BCMVN can also be extended to segmental implementation to reduce the latency for long utterances.

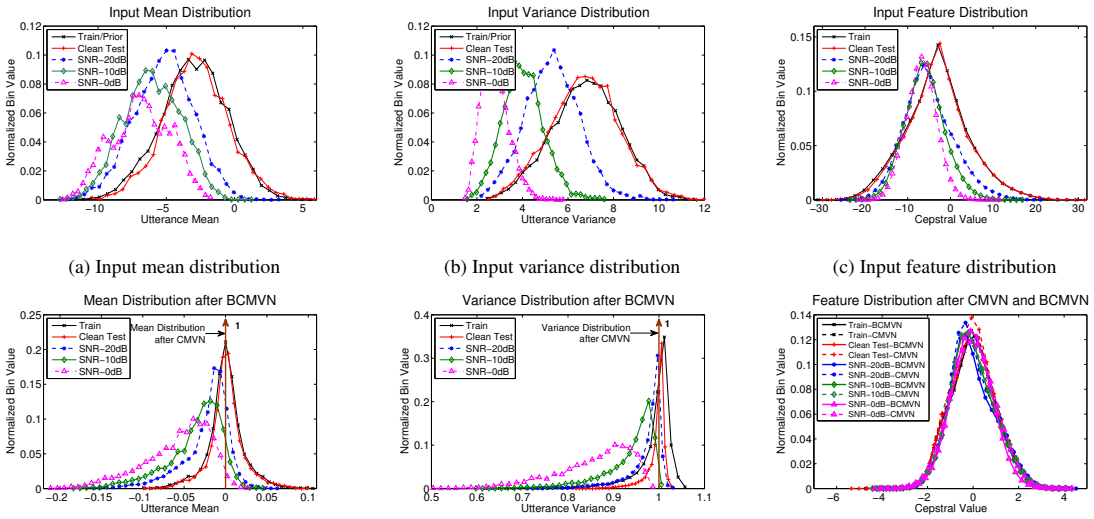
3. ANALYSIS

We next analyze and compare the BCMVN algorithm with CMVN and HEQ, and discuss its computational advantages over HEQ.

3.1. CMVN vs BCMVN

- **Non-zero mean and non-unit variance :** For a given utterance, the mean of every dimension of the CMVN-transformed features is zero, i.e., $\mathbb{E}(\hat{x}) = 0$ and the variance is unity, i.e., $Var(\hat{x}) = 1$, where Var denotes the variance. In case of BCMVN, $\mathbb{E}(\check{x}) \neq 0$ and $Var(\check{x}) \neq 1$.
- **Dimension-specific information :** In the CMVN-transformed features, the mean and the variance of i^{th} and j^{th} dimension are equal for a given utterance, i.e., $\mathbb{E}(\hat{x}(i)) = \mathbb{E}(\hat{x}(j)) = 0$ and $Var(\hat{x}(i)) = Var(\hat{x}(j)) = 1$. In case of BCMVN, $\mathbb{E}(\check{x}(i)) \neq \mathbb{E}(\check{x}(j))$ and $Var(\check{x}(i)) \neq Var(\check{x}(j))$. This is because μ_{post} and σ^2_{post} are a function of the prior parameters ($\mu_0, \kappa_0, \alpha_0, \beta_0$) and the ML estimates μ_{ML},σ^2_{ML} (Eq. (12), Eq. (13)). Each dimension would have its own distinct prior values and ML values.
- **Utterance-specific information :** Figures Fig. 1a and Fig. 1b show the distribution of mean and the distribution of variance of input features respectively. The mean and variance are computed from every utterance. The histogram of this set of mean and variance values are plotted for the training set and also for various SNRs of the test data of Aurora2 database for 2^{nd} cepstral coefficient. Now, if we look at the *distribution of* $\mathbb{E}(\hat{x})$ (distribution of mean after CMVN), it would be an impulse at zero, as all utterances after CMVN would have zero mean. But for BCMVN, the *distribution of* $\mathbb{E}(\check{x})$ has a distribution around zero as shown in Fig. 1d. Similar behavior can be observed with variance distribution as-well (Fig. 1e). The distribution of variance after CMVN is an impulse at one, whereas the distribution has a spread after BCMVN.

  The spread in the case of both mean and variance distribution after BCMVN is because the posterior values (μ_{post}, σ^2_{post}) are computed using utterance length T and the ML estimates (μ_{ML} and σ^2_{ML}), and these parameters vary with every utterance. This spread in mean and variance distribution after BCMVN could correspond to capturing utterance-specific information, which does not happen in CMVN.
- **Short Utterances :** BCMVN compensates for the insufficient data by utilizing the prior information to obtain robust



(a) Input mean distribution (b) Input variance distribution (c) Input feature distribution

(d) Mean distribution after BCMVN (e) Variance distribution after BCMVN (f) Feature distribution after CMVN and BCMVN

Fig. 1: Histogram of means, variances and features of utterances for 2^{nd} cepstral coefficient under different noise conditions for Aurora2 data-set input features (top row) and after BCMVN (bottom row). Figure also shows the effect of CMVN.

estimates. In addition, BCMVN preserves utterance-specific information. Fig. 2 shows the distribution of mean and variance of 2^{nd} cepstral coefficient of 4 digits (1, 2, 6 and 9) taken from the Aurora2 training set after applying BCMVN. Utterances with only the particular digit spoken are taken and the histograms are plotted. Clearly, each digit has a distinct mean and variance distribution even after BCMVN. On the other hand, after CMVN, all the digits would have the same mean (zero) and variance (one) and the distribution is an impulse as shown in the same figure.

- **Noisy Cases :** The Bayesian approach computes robust estimate of the parameters as it uses the prior information. As seen from Fig. 1d and Fig. 1e, BCMVN retains the information in the mean and variance by having a spread even in noisy cases, unlike CMVN.

  The goal of CMVN is to reduce the mismatch between the training utterances and the noisy test utterances. The Fig. 1f shows the histogram of features after BCMVN and CMVN. It can be clearly seen that the histogram of features under noisy cases closely match (overlap) the histogram of the training features for both CMVN and BCMVN. This indicates that the BCMVN is also able to reduce the mismatch between the training and test features similar to CMVN.

3.2. BCMVN vs HEQ

If we look at the distribution of utterance mean after HEQ, it would be an impulse at the reference mean, as all utterances are equalized to have the same cdf. Similarly, the distribution of variance after HEQ would be an impulse at the reference variance. This behavior of HEQ is similar to CMVN and the discussion in the previous section (section 3.1) is applicable to HEQ as well.

3.3. Computational Analysis

Before training, BCMVN estimates the parameters of the Normal-Gamma prior distribution for all the 13 dimensions. This is a one

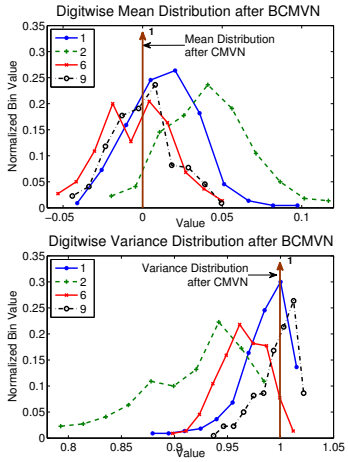


Fig. 2: Digit specific mean and variance distribution of 2^{nd} cepstral coefficient of Aurora2 train dataset after performing BCMVN

time computation and is similar to estimating the reference cumulative distribution function (cdf) for HEQ.

During normalization (training and testing), BCMVN performs only a few additional computations when compared to CMVN, i.e., Eq. (12) and Eq. (13). The order of computation is very minimal for BCMVN when compared with HEQ. For performing Quantile-based HEQ, test cdfs are estimated from the given utterance for all 13 dimensions (involves sorting) and are equalized to the reference pdfs using interpolation.

In summary, the analysis indicates that the BCMVN algorithm achieves the purpose of CMVN by reducing the mismatch between the training and test conditions. In addition, BCMVN captures discriminative information compared to CMVN and HEQ without any additional computational overhead.

Table 2: BCMVN-M recognition results for various γ (% Accuracy)

γ Values	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Aurora2 All	78.34	81.62	**81.75**	81.55	81.46	81.34	81.29	81.26	81.23	81.23
Aurora2 Short	63.75	81.62	83.43	84.14	**84.40**	84.39	84.30	84.25	84.21	84.14
Aurora4	65.00	**65.48**	65.13	64.80	64.79	64.05	63.88	63.50	63.35	63.06

Table 3: Recognition results on Aurora4 database (% Accuracy)

Test Case	01	02	03	04	05	06	07	08	09	10	11	12	13	14	Average
Baseline	90.64	64.17	38.88	44.01	41.36	42.87	38.73	69.79	47.84	31.01	32.73	28.47	33.70	28.40	45.19
CMN	89.78	80.38	56.66	55.58	49.71	58.99	48.98	81.28	67.63	47.43	44.26	36.99	49.47	38.67	57.56
CMVN	90.72	78.98	59.67	59.95	57.74	62.17	54.23	81.49	66.73	49.32	47.49	43.62	50.51	44.39	60.50
HEQ	90.92	79.66	64.62	63.07	62.19	66.49	61.70	81.82	69.16	55.20	52.49	50.31	55.37	51.50	64.61
BCMVN	90.70	79.32	62.94	62.17	61.91	64.80	59.39	82.18	68.50	52.44	49.88	47.51	53.20	47.88	63.06
BCMVN-M	90.73	81.08	66.21	64.06	64.15	67.38	61.82	83.39	71.32	55.54	53.00	51.00	56.04	51.04	**65.48**

4. MODIFIED BCMVN (BCMVN-M)

The posterior estimate can be interpreted as a weighted combination of the prior (μ_0) and the ML estimate (μ_{ML}). For instance, Eq. (12) can be re-written as

$$\mu_{post} = \frac{\kappa_0}{\kappa_0 + T}\mu_0 + \frac{T}{\kappa_0 + T}\mu_{ML} \tag{19}$$

It can be seen that as T increases, more weight is given to the ML estimate. In the limit $T \to \infty$, the posterior estimate (μ_{post}) would converge to the ML estimate. As T increases, the amount of discriminable information captured could decrease for BCMVN as the posterior estimates would start converging towards the ML estimates. If the weight of the prior information can be made higher compared to the weight of the ML estimate, BCMVN would be able to preserve more discriminable information (as discussed in section 3.1) and could improve the recognition performance.

BCMVN can be modified to give more weight to the prior information, by weighting T by γ such that $0 < \gamma < 1$, i.e., $T_w = T \times \gamma$. This modified T_w is then used in Eq. (12) and Eq. (13) instead of T, to estimate the posterior parameters. We denote this approach of modifying BCMVN as *BCMVN-M*.

The value of γ can be empirically chosen for different databases based on the length of the utterances and the recognition performance. Experiments were performed on Aurora2 and Aurora4 databases to study the behavior of BCMVN-M. T was modified by varying γ from 0.1 to 1.0 in steps of 0.1. Table 2 shows the recognition performance of the BCMVN-M for various γ values for the databases. It can be observed that for long utterances (Aurora4) higher performance is obtained with smaller $\gamma = 0.2$, and for short utterances (Aurora2 Short) the optimal γ is at 0.5.

5. EXPERIMENTAL SETUP

Databases: The experiments are conducted on the Aurora2 [13] and the Aurora4 [14] databases distributed by ELRA. The Aurora2 database comprises of connected spoken digits contaminated with different types of noise at various SNR levels. In addition to comparing the performances of entire test data set, we do tests on short utterances as well. Utterances that have a maximum of two spoken digits are considered as short utterances. There are 70070 utterances in the entire test data set inclusive of all noise conditions, out of which 29799 are short utterances (one or two spoken digits).

The Aurora4 database is a continuous speech database built as a dictation task on texts from the Wall Street Journal with a word size of 5000. It comprises of 1 clean training, 7 types of additive noise test cases (test 01 to test 07) and another 7 test cases with both convolutional and additive noise.

Feature Extraction: HMM Toolkit (HTK) 3.4 is used for experiments. Standard MFCC vectors are used for basic feature parametrization. Short time Fourier transform of pre-emphasized speech signal is obtained using $25ms$ window and shift size of $10ms$. 23 Mel-scaled filter banks are used for smoothing the spectrum. 13-dimensional cepstral coefficients are used (inclusive of C_0).

CMN features are obtained by subtracting utterance-wise mean value from each cepstral coefficient of the given utterance. CMVN is also performed utterance-wise as in Eq. (3). HEQ features are obtained by transforming the utterances to match clean speech cdf as done in [5]. Clean speech cdf is obtained from all the training utterances. BCMVN is implemented as described in section 2. Finally for each experiment, 13 delta and 13 acceleration coefficients are appended to get composite 39-dimensional MFCC vector per frame. All four feature normalization techniques are applied on both training data and test data.

Acoustic Modeling: For the Aurora2 database, the acoustic model is a left to right continuous density HMM with 16 states and 3 diagonal covariance Gaussian mixtures per state. Word level HMM model is used. Training is done using clean train utterances from the Aurora2 dataset.

In the case of Aurora4 database, continuous cross-word triphone models with 3 states are used. Each state is modeled using 16 Gaussian mixtures and silence state with 32 mixtures. A total of 3063 tied-states are built. Standard WSJ0 bi-gram language model is used.

6. RESULTS

The performance of the BCMVN algorithm is compared against CMN, CMVN and HEQ for both the Aurora2 and the Aurora4 databases. We also compare the performance on short utterances (refer to section 5) in the Aurora2 database.

Table 3 compares the performances of the algorithms in the Aurora4 database. The recognition rate of the BCMVN is consistently better than CMVN and CMN for the reasons discussed in section 3. In addition, the performance of the BCMVN is comparable to HEQ. BCMVN-M is reported with $\gamma = 0.2$ for which the optimal performance was obtained (Table 2). It can be seen that BCMVN-M has higher recognition performance compared to HEQ. BCMVN-M has a relative WER reduction of 18.7%, 12.6% and 2.8% compared to CMN, CMVN and HEQ respectively.

Table 4: Recognition results on Aurora2 database

(a) % Accuracy - All Utterance

	Baseline	CMN	CMVN	HEQ	BCMVN	BCMVN-M
Clean	99.12	99.25	99.11	99.08	99.16	99.14
SNR20	95.49	97.35	96.99	97.66	97.71	97.88
SNR15	84.85	93.43	93.67	95.54	95.87	96.10
SNR10	60.39	80.62	85.76	90.14	90.82	91.15
SNR5	30.70	51.87	65.96	76.07	76.29	76.68
SNR0	13.24	24.30	32.63	45.32	45.46	46.95
SNR-5	8.15	12.03	12.64	16.70	18.99	19.20
Average	56.93	69.51	75.00	80.94	**81.23**	**81.75**

(b) % Accuracy - Short Utterance

	Baseline	CMN	CMVN	HEQ	BCMVN	BCMVN-M
Clean	99.47	99.49	99.22	98.98	99.33	99.31
SNR20	92.61	98.22	96.44	96.14	97.90	97.95
SNR15	72.98	95.91	92.40	93.37	96.58	96.91
SNR10	32.77	88.54	83.81	85.90	92.47	92.84
SNR5	-5.59	68.60	65.18	71.25	80.40	80.50
SNR0	-10.29	43.77	35.14	41.33	53.35	53.80
SNR-5	-2.58	22.12	14.44	15.59	26.80	24.84
Average	36.49	79.01	74.59	77.60	**84.14**	**84.40**

BCMVN-M All Utterance : $\gamma = 0.3$, BCMVN-M Short Utterance : $\gamma = 0.5$

For the Aurora2 database, it can be seen from Table 4a that the BCMVN works better compared to CMN, CMVN and HEQ for all utterances. For the short utterances (Table (4b)), the performance of HEQ and CMVN decrease compared to CMN for the reasons discussed in section (1.1). BCMVN has higher recognition rate compared to CMN, CMVN and HEQ for both short and all utterances, as BCMVN preserves additional information (section 3) compared to other normalization techniques. The BCMVN has relative WER reduction of 38.6% compared to CMVN and 30.4% compared to HEQ in the case of short utterances. As in the case of Aurora4, BCMVN-M further improves the performance of BCMVN for both short utterances and all utterances. The optimal value for BCMVN-M for all utterances was obtained with $\gamma = 0.3$ and for short utterances the optimal performance was obtained with $\gamma = 0.5$ (Table 2).

7. CONCLUSION AND FUTURE WORK

In this paper we have presented a computationally efficient feature normalization technique, BCMVN - an improved version of CMVN. BCMVN uses the posterior estimates of mean and variance in CMVN instead of the ML estimates. Our analysis indicate that BCMVN captures more discriminable information compared to CMVN. Experiments conducted on the Aurora2 and the Aurora4 databases show that BCMVN outperforms CMN, CMVN and HEQ. BCMVN works for short utterances as well, whereas the performances of HEQ and CMVN degrade. BCMVN has negligible increase in the computational cost when compared to CMVN and is significantly efficient compared to HEQ. The higher recognition performance and computational efficiency makes BCMVN very relevant for IVR-based applications where the utterances could be short and noisy.

BCMVN is currently performed on conventional per-utterance CMVN. This can be extended to segmental CMVN and this work is in progress. Research is also being conducted to extend this technique to other normalization algorithms.

8. REFERENCES

[1] S. Furui, "Cepstral analysis technique for automatic speaker verification," *IEEE Transaction on Acoustics, Speech and Signal Processing*, vol. 29, pp. 254–272, 1981.

[2] Yifan Gong, "Speech recognition in noisy environments: A survey," Tech. Rep., CRIN/ CNRS - INRIA-Lorraine, Nancy, France, Nov. 1994.

[3] Olli Viikki and Kari Laurila, "Cepstral domain segmental feature vector normalization for noise robust speech recognition," *Speech Communication*, vol. 25, no. 1, pp. 133–147, 1998.

[4] F. Hilger and H. Ney, "Quantile based histogram equalization for noise robust large vocabulary speech recognition," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 14, no. 3, pp. 845 – 854, May 2006.

[5] A. de la Torre, A.M. Peinado, J.C. Segura, J.L. Perez-Cordoba, M.C. Benitez, and A.J. Rubio, "Histogram equalization of speech representation for robust speech recognition," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 13, no. 3, pp. 355 – 366, May 2005.

[6] Ole Morten Strand and Andreas Egeberg, "Cepstral mean and variance normalization in the model domain," in *ISCA Tutorial and Research Workshop*, 2004.

[7] Pedro J Moreno, Bhiksha Raj, and Richard M Stern, "A vector Taylor series approach for environment-independent speech recognition," in *Acoustics, Speech, and Signal Processing, ICASSP-96 Proceedings, IEEE International Conference on*, 1996, vol. 2, pp. 733–736.

[8] Sangita Tibrewala and Hynek Hermansky, "Multi-band and adaptation approaches to robust speech recognition," in *Proc. Eurospeech*, 1997, pp. 2619–2622.

[9] Pere Pujol, Dušan Macho, and Climent Nadeu, "On real-time mean-and-variance normalization of speech recognition features," in *Acoustics, Speech and Signal Processing, ICASSP 2006 Proceedings, IEEE International Conference on*. IEEE, 2006, vol. 1, pp. 773–776.

[10] Vikas Joshi, N Vishnu Prasad, and S Umesh, "Modified cepstral mean normalization - transforming to utterance specific non-zero mean," in *Interspeech, Lyon, France*, August 2013, pp. 881–885.

[11] Y. Obuchi and R Stern, "Normalization of time-derivative parameters using histogram equalization," in *Proc. of EUROSPEECH 2003, Geneva, Switzerland*, pp. 665–668.

[12] Kevin P Murphy, "Conjugate Bayesian analysis of the Gaussian distribution," Tech. Rep., University of British Columbia, October 2007.

[13] David Pearce and Hans-Günter Hirsch, "The Aurora experimental framework for the performance evaluation of speech recognition systems under noisy conditions," in *ISCA ITRW ASR2000*, 2000, pp. 29–32.

[14] Günter Hirsch, "Experimental framework for the performance evaluation of speech recognition front-ends on a large vocabulary task," Tech. Rep., Ericsson, Nov. 2002.