Powered Cepstral Normalization (P-CN) for Robust Features in Speech Recognition

Chang-wen Hsu and Lin-shan Lee

Graduate Institute of Communication Engineering
National Taiwan University, Taiwan, Republic of China
poseidons@speech.ee.ntu.edu.tw , lslee@gate.sinica.edu.tw

Abstract

Cepstral normalization has been popularly used as a powerful approach to produce robust features for speech recognition. Good examples of approaches in this family include the well known Cepstral Mean Subtraction (CMS) and Cepstral Mean and Variance Normalization (CMVN), in which either the first or both the first and the second moments of the Mel-frequency Cepstral Coefficients (MFCCs) are normalized. In this paper, an improved approach of Powered Cepstral Normalization (P-CN) is proposed to normalize the MFCC parameters in the *r*-th powered domain, where $r > 1.0$. The basic idea is that when the MFCC parameters are raised to the *r*-th power, the harmful parts of environmental disturbances may be more emphasized than the speech features which are relatively smooth. Therefore performing the normalization in the domain of the *r*-th power may be more helpful. But the value of *r* should not be too large because in that case the environmental disturbances may be exaggerated and further corrupt the speech features. This approach is computationally simple and efficient. Initial experimental results on AURORA 2.0 testing environment showed that significant improvements in recognition rates are consistently obtainable under all different noisy conditions.

Index Terms: Robust speech recognition, cepstral normalization, cepstral mean and variance normalization.

1. Introduction

In real world speech recognition applications, robust features are highly desired in order to offer acceptable recognition performance under various noisy conditions. Mel-frequency cepstral coefficients (MFCCs) have been well accepted as a good choice for speech features with reasonable robustness, and many advanced techniques have been developed based on them. Normalizing the MFCC parameters has been a well-known approach to improve the robustness of the feature parameters. Cepstral Mean Subtraction (CMS) [1] and Cepstral Mean and Variance Normalization (CMVN) [2] have been two commonly used methods in this family, in which either the first or both the first and the second moments of MFCCs are normalized. A possible reason for this is that CMS effectively removes the DC component in cepstral domain, which usually includes the channel distortion, and avoids the low frequency noise to be further amplified. The variance normalization in CMVN, on the other hand, may reduce the mismatch in the statistics of the training and testing speech signals. It was also proposed that additional normalization of the third-order cepstral moment may achieve even better performance [3], because with such normalization the above mismatch may be further reduced. In addition, Histogram Equalization (HEQ) [4] and Higher Order Cepstral Moment Normalization (HOCMN) [5] are two other efficient methods recently proposed for cepstral normalization, and have been shown to offer better performance than the previous ones.

In this paper, we proposed a new approach of Powered Cepstral Normalization (P-CN). The basic idea is that when the MFCC parameters are raised to the *r*-th power, where $r > 1.0$, the harmful parts of environmental disturbances may be more emphasized than the speech features which are relatively smooth. Therefore performing the normalization in the domain of the *r*-th power may be more helpful. But the value of *r* should not be too large because in that case the environmental disturbances may be exaggerated and further corrupt the speech features. Therefore a good value of *r* should be carefully chosen. In the following, the proposed Powered Cepstral Normalization (P-CN) are first formulated in section 2. The experimental setup based on AURORA 2.0 testing environment is described in section 3, and some initial experimental results and discussions are presented in section 4. Finally, we make concluding remarks in section 5.

2. Powered cepstral normalization (P-CN)

2.1. Cepstral normalization

Most cepstral normalization approaches try to normalize the MFCC parameters with respect to the moments, although some other approaches (such as histogram equalization (HEQ)) don't. In order to have a unified formulation, here we first briefly summarize the well-known approaches of CMS and CMVN as two typical examples of cepstral normalization, and then develop the concept of the new approach of powered cepstral normalization (P-CN) based on them. The *N*-th order moment of a MFCC parameter sequence $X(n)$, where *n* is the time index, is the expectation value of $X^N(n) \equiv [X(n)]^N$, usually approximated by the time average over some interval,

$$E[X^N(n)] \triangleq \frac{1}{T}\sum_{k=0}^{T-1} X^N(k) \cdot \quad (1)$$

With the above notation, the well-known CMS processing is

$$CMS[X(n)] = X_{CMS}(n) \triangleq X(n) - E[X^1(n)], \quad (2)$$

and the well-known CMVN processing is

$$CMVN[X(n)] = X_{CMVN}(n) \triangleq X_{CMS}(n) \Big/ \sqrt{E[X^2_{CMS}(n)]} \cdot \quad (3)$$

2.2. Powered cepstral normalization (P-CN)

Here the only additional process for Powered Cepstral Normalization (P-CN) is to raise the MFCC parameters to the *r*-th power, but we need to retain the sign of the original parameters, and only raise the absolute value to the *r*-th power. Then we can perform the same normalization techniques over the powered MFCC parameters as usual, such as CMS or CMVN, and then transform them back to the original MFCC

domain. Let $X(n)$ be the sequence for an original MFCC parameter and n be the time index as before, the transformation from $X(n)$ to the r-th power domain, $Y(n)$, is thus simply

$$P^r[X(n)] = Y(n) \triangleq \mathrm{sgn}[X(n)] \cdot |X(n)|^r . \quad (4)$$

We can then perform the moment normalization procedure as usually on $Y(n)$, for example,

$$Y'(n) = CMS[Y(n)] \text{ or } Y'(n) = CMVN[Y(n)] \quad (5)$$

for CMS and CMVN respectively. We then transform the new feature coefficients back to the original domain,

$$X'(n) = P^{1/r}[Y'(n)] \triangleq \mathrm{sgn}[Y'(n)] \cdot |Y'(n)|^{1/r}, \quad (6)$$

where $P^{1/r}[\cdot]$ can be performed with equation (4) except here r is replaced by $1/r$. In this way we can perform Powered CMS (P-CMS) and Powered CMVN (P-CMVN) easily. Other cepstral normalization approaches, not limited to CMS and CMVN, can be similarly performed in the r-th power domain as well.

2.3. Further discussions about powered cepstral normalization

In the CMS and CMVN cases, the orders of the moments being normalized are always integers (1 or 2). However, when we consider to normalize the cepstral parameters raised to the r-th power as discussed above, where r is a positive real number, a very similar concept is to normalize the moments with orders being a non-integer, say any positive real number u. With such considerations, we can define two types of generalized moments as given below. In the first case, in evaluating the moments the sign of each parameter sample $X(n)$ in equation (1) is retained first and only the absolute value of the sample $X(k)$ is raised to a non-integer power order u. So equation (1) is generalized to

$$E_1\left[X^u(n)\right] \triangleq \frac{1}{T}\sum_{k=0}^{T-1} \mathrm{sgn}\left[X(k)\right] \cdot \left(abs\left[X(k)\right]\right)^u . \quad (7)$$

$E_1[X^u(n)]$ in the above is referred to as the generalized moment of the first type with order u here. In the second case, in evaluating the moments the sign of each parameter sample $X(k)$ in equation (1) is simply removed,

$$E_2\left[X^u(n)\right] \triangleq \frac{1}{T}\sum_{k=0}^{T-1} \left(abs\left[X(k)\right]\right)^u . \quad (8)$$

$E_2[X^u(n)]$ in the above is referred to as the generalized moment of the second type with order u here. With the above definitions, $E_1[X^u(n)]$ in equation (7) reduces to equation (1) if u is an odd integer, and $E_2[X^u(n)]$ in equation (8) also reduces to equation (1) if u is an even integer. So the conventional definition of moments in equation (1) remains valid here for integer orders, in which case the two types in equations (7) and (8) converges into one in equation (1).

When we have the definitions of the two types of generalized moments as defined above, it is interesting to discuss how they are related to the powered cepstral normalization discussed previously. Because the generalized moments of the first and the second types defined in equations (7) (8) can reduce to the conventional moments when the moment orders are odd and even integers respectively, these generalized moments can be used in the discussion. The general idea is summarized in Table 1. In rows (1) and (2) of Table 1, it is easy to see that for the cepstral parameter sequence $X(n)$ the generalized moment of the first type with order 1.0 has been normalized for both CMS and CMVN, and for CMVN the generalized moment of the second type with order 2.0 has also been normalized in addition. For P-CMS mentioned in section 2.2 with order r as listed in row (3),

Generalized Cepstral Normalization		Orders of Generalized Moments being Normalized	
		First Type	Second Type
(1)	CMS	1.0	—
(2)	CMVN	1.0	2.0
(3)	P-CMS	r	—
(4)	P-CMVN	r	$2r$

Table 1. *Comparison for the orders of the two types of generalized moments being normalized for different cepstral normalization techniques discussed here in this paper.*

actually the generalized moment of the first type with order r for the parameter sequence $X(n)$ has been normalized, since the conventional first moment in the domain of the r-th power corresponds to the generalized moment of the first type with order r in the domain of the original parameter sequence $X(n)$. Therefore P-CMS proposed here by raising the cepstral sequence to the r-th power and performing CMS in the r-th power domain can be considered as a simple approach to normalize the generalized moment of the first type with order r for a parameter sequence. Similarly, for P-CMVN also mentioned in section 2.2 with order r as in row (4), everything remains the same as P-CMS, except here the generalized moment of the second type with order $2r$ has been normalized in addition.

3. Experimental setup

The above approaches were evaluated by the AURORA 2.0 testing environment with an English connected-digit string corpus. Two training conditions (clean condition / multi-condition) and three testing sets (sets A/B/C) were defined by AURORA 2.0 [6]. In clean-condition training the acoustic models are trained by clean speech only, while in multi-condition training the models are trained by a corpus with both clean and noisy speech. The testing set A included four different types of noise which were used in the multi-condition training (subway, babble, car and exhibition), while the testing set B included another four different types of noise not used in the multi-condition training (restaurant, street, airport and train station). The testing set C then included two noise types respectively from sets A and B (subway and street), plus additional convolutional noise. Six different SNR values, ranging from 20dB to -5dB, were tested in each case. Whole-word HMM models were used as specified by AURORA 2.0. Each word had 16 states and 3 Gaussian mixtures per state. The speech features were extracted by the AURORA WI007 Front-end, which converted each signal frame into 13 cepstral coefficients (MFCCs, C0~C12), on which all the normalization techniques proposed above were performed. The first and second derivatives were then computed from the normalized cepstral coefficients and used as well in the tests. The P-CN approaches proposed here were tested in a way, in which the cepstral normalization performed in the r-th power domain was based on the statistics of progressively moving segments. In other words, the summation in equation (1) was performed over a progressively moving segment with length l+1, including the preceding l/2 frames and following l/2 frames where l = 140 in our experiments.

3.1. Development set and object function based on the clean-condition training set of AURORA 2.0

In addition to the testing environment as summarized above, we also defined a development set based on this testing

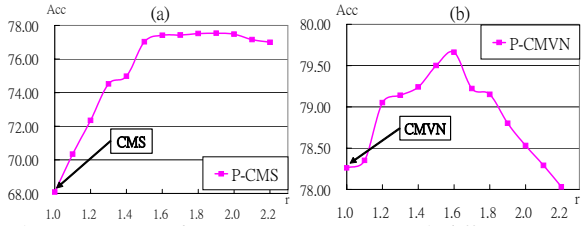

Figure 1 *Averaged recognition accuracy with different power order r for (a) P-CMS and for (b) P-CMVN*

Clean Condition		Set A	Set B	Set C	Avg.	Error Reduction
(1)	CMS	65.81	71.17	66.45	68.08	—
(2)	P-CMS (r = 1.9)	76.67	78.32	77.73	77.54	29.64%

Table 2. *Recognition accuracies for experiments with CMS and the proposed P-CMS under clean-condition training.*

environment, to be used for the selection of the various power orders r as mentioned in section 2.2. For this purpose, we divided all the 8440 utterances in the clean training corpus of AURORA 2.0 into two subsets, 7544 utterances for training and the rest 896 for testing. We added the eight types of noise used in AURORA 2.0 as summarized above on the second subset of 896 utterances (now defined as the testing data of the development set, originally in the clean training corpus of AURORA 2.0) with SNR ranging from 20dB to -5dB respectively as the testing data for the development set. The first subset of 7544 utterances was then used for clean-condition training for the development set. So the testing conditions for the development set is very similar to those with clean-condition training and testing sets A and B defined in AURORA 2.0. The averaged word accuracy for all these forty conditions (eight types of noise and five SNR values) was then used as the object function for parameter selection.

4. Preliminary experimental results

4.1. Experiments for CMS and P-CMS

All the preliminary experiments reported here in this paper were performed with the clean training condition only, because this represents a more serious mismatch situation and requires more robust speech features. The first set of experiments used CMS as the initial example, i.e., comparing the conventional CMS and the proposed P-CMS. The curve in Figure 1 (a) is the recognition accuracy for P-CMS for different values of r averaged over all different SNRs from 20dB to 0dB and all the three testing sets A, B and C with all types of noise. The case r = 1.0 corresponds to the conventional CMS. It can be found from Figure 1 (a) that the recognition accuracy for P-CMS actually monotonically increases with r when r is between 1.0 and about 1.9. The performance then degrades slightly when r is beyond 1.9. Such results are quite reasonable as mentioned previously. Some good value of r may emphasize the disturbances more so that they can be better normalized, while too high value of r may exaggerate the disturbances and further corrupt the speech features. The curve in Figure 1 (a) verified the concept here. The recognition accuracy averaged separately for the three testing sets A, B, and C respectively for all types of noise in each set with all SNR values for P-CMS with the best case of r = 1.9 in Figure 1 (a) are listed in row (2) of Table 2, as compared to the results for corresponding cases for CMS (the point of r = 1.0 in Figure 1 (a)). The improvements are

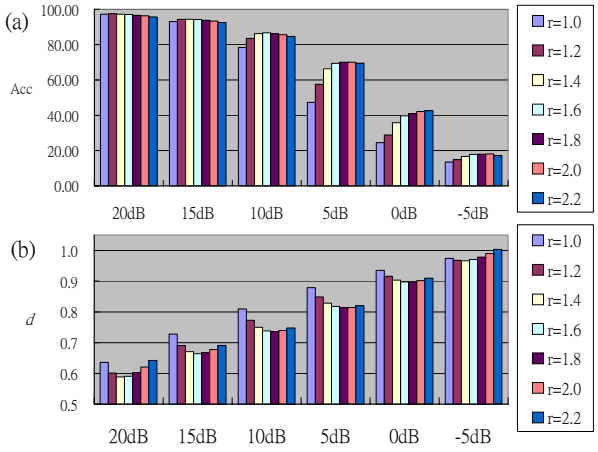

Figure 2 *(a)Performance and (b) averaged distance measure d of P-CMS for several typical values of r for different SNR values, averaged over all different types of noise in different testing sets, but separated for different SNR values.*

consistent across all the sets, the most significant for sets A and C (over 30% error rate reduction), with the overall averaged improved from 68.08% to 77.54%.

4.2. Further analysis comparing CMS and P-CMS

We then compared the performance for several typical values of r in P-CMS averaged over all different types of noise but separated for different SNR values in Figure 2 (a). The first bar in Figure 2 (a) for each case is for r = 1.0, i.e., the case of conventional CMS. Several observations can be made here. First, the improvements obtained with the approaches proposed here for r > 1.0 were quite obvious and consistent across all SNR values. The improvements were especially significant for SNR being 5dB or 0dB. It can be found that the accuracy could be improved from 47.36% for r = 1.0 to 70.02% for r = 2.0 at 5dB of SNR and from 24.44% for r = 1.0 to 42.62% for r = 2.2 at 0dB of SNR. Second, the best values of r were actually SNR dependent. It was roughly 1.2 for clean and 20dB cases, then increases as SNR decreases. For 0dB of SNR, r = 2.2 turned out to be best in Figure 2 (a), but the best value may be beyond 2.2. For -5dB of SNR, the best value of r actually returned to roughly 2.0. Such results are reasonable. With stronger disturbances, a higher power order r may produce a more appropriate domain where the emphasized disturbances can be properly normalized. When the disturbances are too strong, however, such as -5dB of SNR, the disturbances may be too much exaggerated for higher r and can't be normalized well. The best value of 1.9 as obtained previously in Figure 1 (a) was then the result when averaging the accuracy for SNR values from 20dB to 0dB.

The effect of P-CMS on each individual feature vectors can be analyzed with averaged distance measure d defined as

$$d = E\left[\frac{\|\bar{y} - \bar{x}\|}{\|\bar{x}\|}\right] \tag{9}$$

where $\bar{x}$ is the 13-dimensional vector of MFCC parameters for clean speech but normalized by the conventional CMS (r = 1.0), and $\bar{y}$ is the corresponding noisy speech version processed by P-CMS (with a specific power order r) and normalized by the variance of the clean speech feature parameters processed by P-

Clean Condition	Set A	Set B	Set C	Avg.	Error Reduction
(1) CMVN	77.52	78.86	78.53	78.26	—
(2) P-CMVN (r = 1.6)	79.03	80.11	80.05	79.67	6.49%
(3) Improved P-CMVN	80.49	82.72	80.59	81.40	14.44%

Table 3. *Recognition accuracis for experiments with CMVN and the proposed P-CMVN under clean-condition training.*

CMS (with the same power order r). The various normalization processes performed on $\bar{x}$ and $\bar{y}$ as mentioned above were to establish a reasonable common base for distance evaluation. The results of the averaged distance measure d for different values of r for different SNR values are shown in Figure 2 (b).

By comparing the averaged distance measure in Figure 2 (b) with the recognition accuracies in Figure 2 (a), very high correlation between the results in these two figures can be found, i.e., smaller distance in Figure 2 (b) implied the P-CMS-processed ($r > 1.0$) noisy feature vectors were better matched to the clean speech feature vectors "individually" as compared to the conventional CMS ($r = 1.0$). The smaller distance in Figure 2 (b) is almost directly related to higher recognition accuracy in Figure 2 (a). In figure 2 (b) for each SNR values, increasing the value of r from 1.0 in general reduced the averaged distance, although the distance may be increased if the value of r was too large. Smaller distance measures were obtained for higher SNR values. All these are in agreement with the accuracies in Figure 2 (a) to a good extent. These results explained the effectiveness of the P-CMS approach proposed here.

4.3. Experiments for CMVN and P-CMVN

The experiments of P-CN with CMVN were similarly performed, and the results are in Figure 1 (b). Figure 1 (b) is in parallel with Figure 1 (a), in which the performance of P-CMVN (with fixed value of r) averaged over all types of noise and all SNR values are plotted as functions of r ranging from 1.0 up to 2.2. The results here are very similar to those in Figure 1 (a), except here the best performance is obtained when r is about 1.6 for P-CMVN.

In the formulation in section 2.2, we in general assume the same value of r is used for all the (13 or so) different MFCC parameters, but of course this is a simplified assumption. Apparently the best values of r may be different for different MFCC parameters. With the above concept, the best values of r for each MFCC parameter can at least be approximated using a simple greedy algorithm, in which the value of r is adjusted step by step for each MFCC parameter with the development set and object function as defined in section 3.1 for optimization. The P-CN obtained in this way is referred to as Improved P-CN here.

The results for CMVN (baseline), P-CMVN (with the best value of r = 1.6) and Improved P-CMVN (optimized with individual MFCC parameters) are respectively listed in rows (1), (2) and (3) of Table 3, each separately for sets A, B and C while averaged over all types of noise within the sets and all SNR values. Similar to Table 2, significant improvements can be found in rows (2) and (3) of Table 3 as compared to row (1). We further compared the performance of CMVN and Improved P-CMVN (rows (1) and (3) in Table 3) for different SNR values (averaged over all types of noise) in Figure 3 (a) and different types of noise (averaged over all SNR values) in Figure 3 (b) respectively, and we can find that Improved P-CMVN is consistently better than CMVN in all cases including higher SNR conditions. From Figure 3 (b), we can see the Improved P-CMVN performs significantly better than the conventional

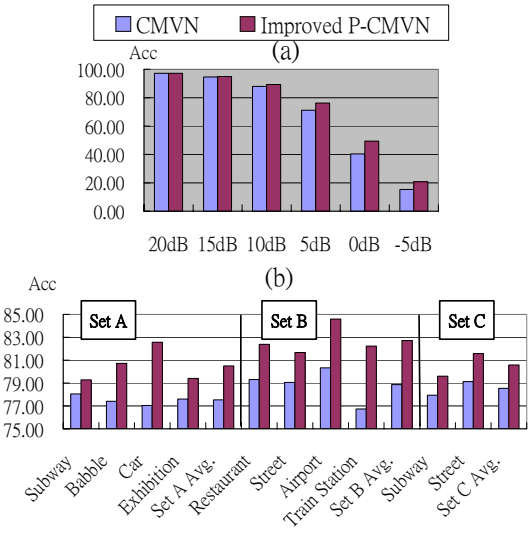

Figure 3 *Comparison of CMVN and Improved P-CMVN for (a) different SNR averaged over all types of noise and (b) different types of noise averaged over all SNR values.*

CMVN for both car noise (very stationary) in set A and train station noise (very non-stationary) in set B with 20.39% and 18.25% error rate reduction respectively, as two typical examples.

5. Conclusions

In this paper, we proposed a new approach for powered cepstral normalization to produce robust features for speech recognition. Experimental results with AURORA 2.0 testing environment verified that significant improvements are consistently achievable in all cases especially under highly mismatched conditions.

6. References

[1] S. Furui, "Cepstral Analysis Technique for Automatic Speaker Verification", IEEE Trans. on ASSP, 1981.

[2] O. Viikki, K. Laurila, "Cepstral Domain Segmental Feature Vector Normalization for Noise Robust Speech Recognition", Speech Communication, Vol. 25, pp. 133-147, August 1998.

[3] Y. H. Suk, S. H. Choi, H. S. Lee, "Cepstrum Third-order Normalization Method for Noisy Speech Recognition", Electronics Letters, Vol. 35, no. 7, pp. 527-528, April 1999.

[4] Á. de la Torre, J. C. Segura, C. Benítez, A. M. Peinado, and A. J. Rubio, "Non-linear Transformations of the Feature Space for Robust Speech Recognition", *ICASSP'02*, 2002.

[5] Chang-wen. Hsu, Lin-shan Lee, "Higher Order Cepstral Moment Normalization (HOCMN) for Robust Speech Recognition", *ICASSP'04*, 2004.

[6] H. G. Hirsch, D. Pearce, "The AURORA Experimental Framework for the Performance Evaluations of Speech Recognition Systems under Noisy Conditions", ISCA ITRW ASR2000, Paris, September 2000.