Extended Powered Cepstral Normalization (P-CN) with Range Equalization for Robust Features in Speech Recognition

Chang-wen Hsu, Lin-shan Lee
Graduate Institute of Communication Engineering
National Taiwan University, Taiwan, Republic of China
posidons@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. A new approach of Powered Cepstral Normalization (P-CN) was recently proposed to normalize the MFCC parameters in the $r$-th order powered domain, where $r_1 > 1.0$, and then transform the features back by a $1/r_2$ power order to a better recognition domain, and it was shown to produce robust features. Here we further extend P-CN to a more effective and efficient form, in which we can on-line find good values of $r_2$ for each utterance in real time based on the concept of dynamic range equalization. The basic idea is that the difference in dynamic ranges of feature parameters is in fact a good indicator for the mismatch degrading the recognition performance. Extensive experimental results showed that the Extended P-CN with range equalization proposed in this paper significantly outperforms the conventional Cepstral Normalization and P-CN in all noisy conditions.

Index Terms: robust speech recognition, cepstral normalization, cepstral mean and variance normalization

1. Introduction

Cepstral Mean Subtraction (CMS) and Cepstral Mean and Variance Normalization (CMVN) have been popularly used to derive relatively robust features in speech recognition [1, 2]. They simply normalize the first and/or the second moment of the cepstral features. Both of them are easy to implement and can effectively reduce the mismatch between training and testing feature parameters to improve the recognition performance. Histogram Equalization (HEQ) was also proposed and popularly used with further improved performance, in which the entire distribution of the cepstral features is equalized, or effectively moments of all orders are normalized [3]. A new technique of Higher Order Cepstral Moment Normalization (HOCMN) was also proposed, in which a few cepstral moments of higher order $N$ ($N > 1$ or 2) are normalized, and shown to provide very good performance [4-5]. Another new approach of Powered Cepstral Normalization (P-CN) was then recently proposed to perform similar cepstral moment normalization, but on a raised power domain. They are almost equally easy to implement, and was shown to be able to significantly improve the robustness of speech feature parameters as well [6]. In this paper, an extended Powered Cepstral Normalization (P-CN) with range equalization is proposed, and is shown to be able to offer further improvements.

We start with a brief review of the formulation of Powered Cepstral Normalization (P-CN) for the development below. The $N$-th order moment of a cepstral feature sequence $X(n)$, where $n$ is the time index, is the expectation value of $X(n)^N$, usually approximated by the time average over some interval,

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

where $\{ k = 0, 1, \ldots, T-1 \}$ is the time interval over which the average is taken. With the above notation, the well-known CMS processing is

$$CMS[X(n)] = X_{CMS}(n) = X(n) - E[X'(n)],$$

the well-known CMVN processing is

$$CMVN[X(n)] = X_{CMVN}(n) = X_{CMS}(n)/\sqrt{E[X_{CMS}^2(n)]},$$

and this can be further extended to Higher Order Cepstral Moment Normalization (HOCMN) [4-5].

The Powered Cepstral Normalization (P-CN), on the other hand, is to normalize the cepstral moments on a raised power domain. In other words, we raise the cepstral features to the $r$-th power, $r$ can be any positive real number, and perform the cepstral normalization over there. In order to do this, we need first to retain the sign of the original parameters, and only raise the absolute value of them to the $r_1$-th power,

$$P^r[X(n)] = Y(n) = \text{sgn}[X(n)] \cdot |X(n)|^r,$$

where $r_1$ is the power order used here. Then we can perform any cepstral normalization technique mentioned above on the raised powered parameters $Y(n)$,

$$Z(n) = F[Y(n)],$$

where $F[\cdot]$ can be CMS, CMVN, or other normalizations such as HEQ or those with respect to higher order moments. Finally we can transform $Z(n)$ back to the recognition domain, or the domain where the recognition is actually performed,

$$X'(n) = P^{1/r_2}[Z(n)] = \text{sgn}[Z(n)] \cdot |Z(n)|^{1/r_2},$$

where $P^{1/r_2}[\cdot]$ is performed in exactly the same way as equation (5) except here $r_1$ is replaced by $1/r_2$. Most cases $r_2 = r_1$, but it is possible to find a different $r_2$ giving a different domain where speech features are more discriminative. The function $F[\cdot]$ in equation (5) can be any type of normalization. If it is CMS, this is powered CMS (P-CMS). If it is CMVN, this is powered CMVN (P-CMVN), etc. Although in this paper below we only use CMVN as an example to discuss the proposed Extended P-CN approach. The basic idea of P-CN is that when the cepstral features are raised to the $r_1$-th power, $r_1 > 1.0$, the harmful parts of environmental disturbances may be more emphasized than the speech features which are relatively smooth. Therefore normalization of moments in that domain can be more helpful. The value of $r_2$ in equation (6) can also be properly chosen such that too large values can be compressed and the residual mismatch after the function $F[\cdot]$ in equation (5) may be suppressed. However, the value of $r_1$ or $r_2$ should be properly chosen, because with a too large value of $r_1$ the environmental disturbances may be too much exaggerated and further corrupt the speech features, and with a too larger value of $r_2$
the dynamic range of speech features may be over-
compressed. An improved version of P-CN (Improved P-CN) 
was also proposed to find the optimal values of \( r_1 \) and \( r_2 \) \([6]\) 
for each cepstral feature by a development set and object 
function (but \( r_1 = r_2 \) assumed in the previous experiments).

Here in this paper we propose a new approach to find 
better values of \( r_2 \) of P-CN by equalizing the dynamic range of 
the cepstral features, so that the feature coefficients could 
be transformed to a better recognition domain. This is in good 
consistency with some earlier findings \([7]\). The basic idea is 
that the difference in dynamic ranges is in fact a good 
indicator for the mismatch which degrades the recognition 
performance. In this way, good values of \( r_2 \) can actually be 
estimated on-line in real time, and this new approach is 
shown to be able to offer very good improvements for all 
types of noise and SNR values. All the experiments reported 
here were performed on the AURORA 2 testing corpus and 
environment.

In the following, the concept of the Extended P-CN with 
Range Equalization is formulated in section 2. The 
experimental setup based on AURORA 2 testing environment 
is described in section 3, and the preliminary experimental 
results and discussions are presented in section 4. The 
concluding remarks are finally given in section 5.

![Figure 1. Range distributions of C0 for the AURORA 2 testing set in the powered domain processed by CMVN for different SNR values.](image)

2. Extended Powered Cepstral Normalization (P-CN) with Range Equalization

The proposed approach is briefly summarized in this section.

2.1. Range Analysis for Cepstral Features

Figure 1 is an example of the range distributions of C0 
coefficient for the AURORA testing set, raised to power \( r_1 \) by 
equation (4) where \( r_1 \) was chosen by the development set as 
mentioned previously \([6]\), then processed by CMVN as an 
example of the function \( \hat{f}(\cdot) \) in equation (5). Each curve in 
the figure represents the dynamic range of an utterance in the 
AURORA testing set under different SNR values. For each 
SNR value the mean and standard deviation values are also 
marked in the figure. It can be found that the dynamic range 
and its variance both become larger for lower SNR values. 
Such residual mismatch may be a source for the degraded 
recognition performance, as was pointed out previously \([7]\). 
This leads to the concept of properly using equation (6) for P-CN 
to reduce such dynamic range mismatch to transform the 
cepstral features to a better recognition domain. Also from 
the curves in Figure 1 it seems this dynamic range is quite 
different for different utterances. Thus it will be desired if the 
value of \( r_2 \) can be chosen on-line in real time for each utterance.

2.2. Extended P-CN with Range Equalization

In the previous work \([6]\), we developed Improved P-CN 
which chose the value of \( r_2 \) by a pre-defined development set 
and object function. Such an approach is time-consuming 
and requires a development set properly describing the task 
domain. Here in this paper, we use the same idea of choosing 
the power order \( r_2 \) in equation (6) to transform the features to 
a better recognition domain, but based on equalizing the 
dynamic ranges for each utterance on-line in real time rather 
than a development set.

The dynamic range \( R \) of a power normalized cepstral 
feature sequence \( X'(n) \), which has been raised to power order 
\( r_1 \), normalized by \( \hat{f}(\cdot) \) and transformed back by another power 
order \( r_2 \) as in equations (4) (5) (6), is the difference between 
its maximum and minimum value, or the sum of the lengths 
of its positive and negative tails,

\[
R = P^{r_1} [Z_r] + P^{r_2} [Z_r],
\]

where \( P^{r_1} [\cdot] \) is defined as in equation (6), \( Z_r \) and \( Z_r \) 
are respectively the positive and negative tails or the magnitude 
of the maximum and minimum values of the values of \( Z(n) \) 
in equation (5) in the time interval being considered,

\[
Z_r = \max_{n=0}^{\infty} |Z(n)|, \quad Z_r = \min_{n=0}^{\infty} |Z(n)|,
\]

where \( \{n = 0, 1, \ldots, L-1\} \) is the time interval considered for 
range equalization, for example an utterance. The purpose of 
the range equalization here is then to find a best value of \( r_2 \) 
so that the range \( R \) in equation (7) can be equalized to a pre-
defined value, \( R_0 \), which can be derived from clean training 
features,

\[
R = P^{r_1} [Z_r] + P^{r_2} [Z_r] = R_0.
\]

The above requirement can be easily achieved with the 
gradient descent algorithm. For the initial value of \( r_2 \), \( r_2^{(0)} \), 
we can assume first the positive and negative tails contribute 
equally to \( R_0 \), which requires two different values of \( r_2 \), \( r_1 \), 
and \( r_2^{(0)} \).

\[
P^{r_1} [Z_r] = P^{r_1} [Z_r] = R_0 / 2.
\]

The initial value of \( r_2 \), \( r_2^{(0)} \), can then be taken as the average 
of \( r_1 \) and \( r_2 \) solved from equation (10),

\[
r_2^{(0)} = \frac{r_1 + r_2}{2} = \frac{\ln(Z_r) + \ln(Z_r)}{2 \ln(R_0 / 2)}.
\]

The value of \( r_2 \) can then be updated by

\[
r_2^{(e+1)} = r_2^{(e)} + \Delta r_2^{(e)},
\]

where \( \Delta r_2^{(e)} \) can be estimated by

\[
R_0 - (P^{r_1} [Z_r] + P^{r_2} [Z_r]) = \int_{r_2^{(e)}}^{r_2^{(e+1)}} \frac{d(P^{r_1} [Z_r])}{dr} dr + \int_{r_2^{(e)}}^{r_2^{(e+1)}} \frac{d(P^{r_2} [Z_r])}{dr} dr \Delta r_2^{(e)}.
\]

which gives

\[
\Delta r_2^{(e)} = \frac{R_0 - (P^{r_1} [Z_r] + P^{r_2} [Z_r])}{\left( P^{r_1} [Z_r] \ln(Z_r) + P^{r_2} [Z_r] \ln(Z_r) \right)_{r_2^{(e)}}^{r_2^{(e+1)}}}.
\]

This procedure can be continued iteratively,

\[
r_2^{(e+1)} = r_2^{(e)} + \Delta r_2^{(e-1)},
\]

where

\[
\Delta r_2^{(e-1)} = \frac{R_0 - (P^{r_1} [Z_r] + P^{r_2} [Z_r])}{\left( P^{r_1} [Z_r] \ln(Z_r) + P^{r_2} [Z_r] \ln(Z_r) \right)_{r_2^{(e-1)}}^{r_2^{(e-1)+1}}}.
\]

In most cases, only a few iterations are enough for this 
algorithm to converge and the computation cost is low, so
good values of $r_2$ can actually be obtained on-line in real time for each utterance or so.

One implementation issue here is that if the positive and negative tails are close to 1 but asymmetric (e.g., one is slightly larger than 1 and the other is slightly smaller than 1), the denominator of equation (15) may be close to zero, and the value of $\Delta r_1^{\text{new}}$ may diverge and degrade the performance. This can be easily avoided by properly scaling the cepstral features by a chosen factor after performing the normalization function $F[l]$ in equation (5) before performing the gradient descent algorithm in equations (7)-(15), to make sure both the positive and negative tail lengths to be larger than 1. For example, in the preliminary experiments reported below in which the normalization function $F[l]$ is CMVN, we simply normalized the standard deviation to 3 instead of 1 and the results turned out to be good.

2.3. Extended P-CN with Dynamic Desired Range

In the above, we assume a fixed desired range $R_0$, which is the mean value derived from the clean training data. However, practically the value of $R_0$ in equation (9), if evaluated for each utterance, is very often related to the length of the utterance. In general, the dynamic range is larger if more speech frames are included in the utterance considered. Figure 2 is the range distribution of $C_0$ coefficients for each utterance in the clean training set of AURORA 2 in the power domain processed by CMVN as a function of the number of frames in the utterances. This figure verifies the above statement that the range is related to the utterance length. A 2nd order linear regression model can therefore be used to find a fitting curve with respect to these sample points in Figure 2, and the desired range $R_0$ in equations (9)-(15) can be dynamically decided utterance-by-utterance using this curve. This is referred to as Extended P-CN with dynamic desired range here in this paper.

3. Experimental Setup

The above approaches were evaluated by the AURORA 2 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 [8]. 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 (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. Each word had 16 states and 3 Gaussian mixtures per state. The speech features were extracted by the AURORA W007 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. Only CMVN was performed as an example of the normalization function $F[l]$ in equation (5) using a progressively moving window of length $l = 141$. In other words, the summation in equation (1) for evaluation of the first two moments was performed over a progressively moving window including the preceding $l/2$ frames and following $l/2$ frames and $l = 140$.

4. Experimental Results

4.1. Baseline results

All the 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 sets of experiments used the conventional CMVN and the recently proposed P-CMVN ($r_1 = r_2 = 1.6$) and Improved P-CMVN (with $r_1 = r_2$ obtained from a development set and an objective function) [6] as the baselines. The recognition accuracy averaged separately for the three testing sets A, B, and C over all types of noise in each set and all SNR values are listed respectively in the first three rows (1) (2) (3) of Table 1. Very good step-by-step performance improvements under various noisy conditions can be observed. In particular, Improved P-CMVN outperforms the other two approaches for all the three testing sets with 81.40% averaged word accuracy. It achieves 14.46% error rate reduction compared to the conventional CMVN.

4.2. Extended P-CMVN with range equalization

The performance for the Extended P-CMVN with range equalization proposed in this paper as discussed in section 2.2, and that with dynamic desired range as in section 2.3 (denoted as Extended P-CMVN (D)), are listed in rows (4) (5) of Table 1 respectively. The values of $r_1$ used for Extended P-CMVN in equation (4) are exactly the same as those used for Improved P-CMVN [6], but the value of $r_2$ was chosen for each utterance using the approaches proposed here. Apparently, consistent and significant step-by-step improvements can be observed for Extended P-CMVN and P-CMVN (D) in rows (4) (5) for all the three testing sets with 6.39% and 10.53% error rate reduction compared to the Improved P-CMVN in row (3), and 19.92% and 23.47% error rate reduction compared to the conventional CMVN in row (1). Apparently choosing a good value of $r_2$ for each utterance...
is much better than using a constant value for all utterances. We also compared our results with the popularly used HEQ using a progressively moving window of length \( l = 98 \) (this length has been optimized) in row (6) of Table 1. It can be found that HEQ \((l = 98)\) is very close to the proposed approach here when averaged over testing sets A, B and C.

Figure 3 (a) is the detailed recognition accuracies for the six experiments listed in rows (1)-(6) of Table 1, averaged over all different types of noise but separated for different SNR values. The six bars in each set are exactly the three baseline approaches (1) CMVN, (2) P-CMVN, (3) Improved P-CMVN, and the two new approaches proposed here, (4) Extended P-CMVN and (5) Extended P-CMVN (D), as well as the popularly used approach (6) HEQ \((l = 98)\) as in Table 1. It can be found that P-CMVN (the second bar) may suffer slight performance degradation for higher SNR values, but the new approaches proposed here (the fourth and the fifth bars) are apparently better in all cases except with a minor degradation for clean speech as the only exception. However, in general the step-by-step improvements from left to right are obvious.

Table 2 lists the detailed accuracies for the results in Figure 3 (a). The six rows are exactly the six bars in Figure 3 (a). It is now clearer that P-CMVN in row (2) and Improved P-CMVN in row (3) performed very well for lower SNR cases but had less improvements or even slight degradation for higher SNR cases. This is because of the fixed \( r_2 \) value obtained with the development set and object function, which apparently emphasized the lower SNR cases with more improvements. However, Extended P-CMVN in rows (4) and (5) of the table can estimate good values of \( r_2 \) for all utterances and all noise conditions. In fact, significant and consistent improvements can be obtained for almost all SNR values, though there is still a minor degradation for the clean speech case.

When comparing Extended P-CMVN (D) with HEQ \((l = 98)\) (the last two bars in Figure 3 (a) and the last two rows in Table 2), we see the proposed approach is slightly better than HEQ for higher SNR cases (from clean to 10dB). However, HEQ performs very well under very noisy conditions such as 0dB and -5dB.

Figure 3 (b) is similar detailed recognition accuracies for the six experiments listed in rows (1)-(6) in Table 1, except averaged over all SNR values but separated for different types of noise. Significant improvements can be observed here for all types of noise even for the wide band noise (ex: subway noise in set A). It can be observed that there were no improvements from P-CMVN \((2^{nd} \text{bar})\) to Improved P-CMVN (third bar) for the subway noise case because the values of \( r_2 \) were fitted to narrow band noise. However, in the Extend P-CMVN proposed here (the fourth and the fifth bars) the values of \( r_2 \) were chosen for each utterance, therefore suitable to any kinds of noise. When comparing the proposed approach with HEQ \((l = 98)\), here HEQ is slightly better for some types of noise but the proposed approach is slightly better in other types of noise.

5. Conclusions

In this paper, we extend the previously proposed concept of Powered Cepstral Normalization (P-CN) to a more generalized and flexible form by equalizing the dynamic range in addition. Extensive experimental results indicated that this proposed Extended P-CMVN with equalized range can perform significantly better than the conventional CMVN and the previously proposed P-CMVN and Improved P-CMVN for all types of noise, and for both higher and lower SNR values. The computational cost of estimating the parameter here is very low, so that on-line normalization is practically feasible.

6. References


<table>
<thead>
<tr>
<th>Clean Condition</th>
<th>Clean</th>
<th>2dB</th>
<th>4dB</th>
<th>6dB</th>
<th>8dB</th>
<th>10dB</th>
<th>12dB</th>
<th>14dB</th>
<th>16dB</th>
<th>18dB</th>
<th>20dB</th>
<th>22dB</th>
<th>24dB</th>
<th>26dB</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMVN ((r_1 = r_2 = 1))</td>
<td>99.13</td>
<td>97.10</td>
<td>96.62</td>
<td>95.95</td>
<td>94.16</td>
<td>88.57</td>
<td>15.85</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P-CMVN (r_1 = r_2 = 1.6)</td>
<td>98.71</td>
<td>96.55</td>
<td>94.06</td>
<td>87.99</td>
<td>73.72</td>
<td>46.40</td>
<td>19.11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Improved P-CMVN</td>
<td>98.81</td>
<td>97.15</td>
<td>94.65</td>
<td>88.98</td>
<td>76.31</td>
<td>49.38</td>
<td>20.79</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Extended P-CMVN</td>
<td>98.90</td>
<td>97.32</td>
<td>95.56</td>
<td>90.53</td>
<td>78.14</td>
<td>51.32</td>
<td>20.64</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Extended P-CMVN(D)</td>
<td>98.94</td>
<td>97.64</td>
<td>95.77</td>
<td>90.00</td>
<td>79.15</td>
<td>53.56</td>
<td>22.52</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HEQ (l = 98)</td>
<td>98.86</td>
<td>96.98</td>
<td>94.99</td>
<td>90.27</td>
<td>78.93</td>
<td>55.71</td>
<td>26.73</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2. Exact Recognition accuracies listed for the results in Figure 3 (a) for different SNR values but averaged over all different types of noise.