Joint Bayesian Predictive Classification and Parallel Model Combination with Prior Scaling for Robust ASR

Svein Gunnar Petersen

Department of Electronics and Telecommunications
Norwegian University of Science and Technology
sveingun@iet.ntnu.no

Abstract

This paper presents a model compensation approach based on Bayesian predictive classification (BPC). In order to obtain effective prior distributions for BPC, our approach uses parallel model combination (PMC) to set the prior mean, and a likelihood ratio to set a scaled frame-specific prior variance. Experiments on the Aurora 2 database show that the proposed approach results in improved average performance compared to PMC.

Index Terms: Speech recognition, robustness, model compensation

1. Introduction

It is well known that the performance of automatic speech recognition (ASR) systems degrades rapidly in the presence of additive background noise. However, many possible applications for ASR require the systems to maintain an acceptable level of performance under noisy conditions in order to be useful in practice. Thus, noise robustness is an important issue for ASR.

In previous work [1] we presented an approach which combines Bayesian predictive classification [2] (BPC) and parallel model combination (PMC) [3]. This was done by using the knowledge about the parameters of noisy speech obtained by PMC in the BPC prior distributions. The effect of this approach was an increase of the variances of the compensated models, resulting in improved performance at low SNRs compared to PMC for most noise types.

However, in the case of babble noise there was a significant drop in performance compared to PMC. This can be explained as follows. The noise model for babble noise can be viewed as model for “background speech”. When applying PMC with this noise model, the compensated HMMs will model a mixture of speech from different sources. Using these models for recognition will result in more insertion errors compared to the original models, since the compensated models are more likely to find a match in segments containing only babble noise. The use of BPC and the resulting increased variances will make this effect even more pronounced, and an even higher number of insertion errors will occur. In order to avoid this negative effect, some measures must be taken to reduce the number of insertion errors.

In this paper we improve our approach by using a prior that is scaled on a frame-by-frame basis in order to avoid an increase of the variance in periods where we are relatively certain that no speech is present. This is done by calculating a likelihood ratio using a model for noisy speech and a noise model at each frame. Then, the width of the BPC prior is scaled according to the value of the likelihood ratio. The resulting approach can be viewed as compromise between using plain PMC and the joint BPC-PMC approach. As we shall see, the resulting performance is increased for noise types containing background speech at the cost of a performance reduction for other noise types.

2. Parallel Model Combination

Parallel model combination is a technique that combines a HMM for clean speech with a HMM for noise resulting in a HMM that models noisy speech [3]. In this paper, we will assume that the noise model is a single Gaussian, with parameters estimated from the first N frames of each input file.

PMC finds the parameters of the adapted acoustic model by combining each of the states in the clean speech HMMs with the parameters of the noise model. For a given speech state, cepstral domain model parameters for both speech and noise are mapped back to the linear-spectral domain and combined to find parameters for noisy speech1. Then, the resulting parameters are transformed back to the cepstral domain.

In this section we will denote the clean speech cepstral domain parameters by \( \{ \mu_s, \Sigma_s \} \), where \( \mu_s \) is the mean and \( \Sigma_s \) is the covariance matrix. Corresponding parameters for noise and noisy speech are denoted by \( \{ \mu_n, \Sigma_n \} \) and \( \{ \mu_y, \Sigma_y \} \) respectively. Let us first consider a pair of general cepstral domain parameters \( \{ \mu, \Sigma \} \), which could be for either clean speech or noise. The parameters are transformed back to the log-spectral domain by

\[
\mu^\log = C^{-1} \mu, \\
\Sigma^\log = C^{-1} \Sigma (C^{-1})^T,
\]

where \( C \) denotes the DCT matrix. The parameters are then transformed to the linear-spectral domain by

\[
\mu^\lim = \exp(\mu^\log + \Sigma^\log / 2), \\
\Sigma^\lim_{ij} = \mu^\lim_i \mu^\lim_j \left[ \exp(\Sigma^\log_{ij}) - 1 \right]
\]

where the subscripts \( i \) and \( j \) are used to indicate row and column numbers in the mean vectors and covariance matrices. Since speech and noise distributions are usually modeled as Gaussians in the cepstral and log-spectral domains, the corresponding distributions in the linear-spectral domain are assumed log-normal. Then, assuming that the distribution of the sum of two

---

1Combination in the log-spectral domain is also possible by using the so-called log add approximation, but this will not be considered here.

---

This work is done as a part of the BRAGE project, which is organized under the language technology programme KUNSTI and funded by the Norwegian Research Council.
Bayesian predictive density replaces the standard pdf for a mixture of the noisy speech distribution, i.e.,

\[ \mu_y = \mu_y^{\text{lin}} + \mu_y^{\text{dif}} \]

\[ \Sigma_y = \Sigma_y^{\text{lin}} + \Sigma_y^{\text{dif}}. \]

Then, in order to transform the noisy speech parameters back to the log-spectral domain, the inverses of (3) and (4) are used. Finally, the cepstrum domain noisy speech parameters are obtained by taking the DCT:

\[ \mu_y = C \mu_y^{\text{log}} \]

\[ \Sigma_y = C \Sigma_y^{\text{log}} C^T. \]

This compensation method works for static parameters. A similar procedure can be used for delta-parameters. See [4] for details.

3. Bayesian Predictive Classification

Let \( W \) denote a word (or a sequence of words), and let \( Y \) denote a sequence of feature vectors. Most speech recognizers model \( p(W, Y) \) as a parametric distribution \( p_{\Lambda, \Gamma}(W, Y) \), where \( \Lambda \) and \( \Gamma \) denote the parameters. Then, decoding is done based on the plug-in MAP decision rule

\[ \hat{W} = \arg \max_W p_X(W)p_{\Lambda, \Gamma}(W, Y), \]

where \( \Lambda, \Gamma \) are parameter estimates for \( \Lambda, \Gamma \) that have been estimated from training data. In practice this decision rule is suboptimal. There are several reasons for this. First of all, the assumed parametric form of \( p(\Lambda, \Gamma) \) is suboptimal. There are several reasons for this. First of all, the

\[ p(\Lambda, \Gamma) \]

is a parametric distribution

\[ p(\Lambda, \Gamma) = \prod_{m=1}^{M} p(\mu_m, \Sigma_m, D) \]

where \( \Phi = \{\mu_m, \Sigma_m, D : m = 1, \ldots, M\} \) are the parameters of the state, and \( D \) is the dimension of the feature vector. In the following, we will only consider uncertainty in the means. For each state and mixture component we assume that a prior distribution \( p(\mu_m | \phi_m) \) is available, where \( \phi_m \) denotes the hyperparameters. Then, we can calculate the probability density for each state as

\[ \tilde{p}_i(y) = \sum_{m=1}^{M} w_{im} \prod_{d=1}^{D} \int p_{\text{lin}}(y_d | \mu_m) p(\mu_m | \phi_m) d\mu_m \]

\[ = \sum_{m=1}^{M} w_{im} \prod_{d=1}^{D} \tilde{p}_{\text{lin}}(y_d), \]

where we have defined

\[ \tilde{p}_{\text{lin}}(y_d) = \int p_{\text{lin}}(y_d | \mu_m) p(\mu_m | \phi_m) d\mu_m. \]

4. Joint BPC and PMC with Prior Scaling

For the BPC approach to be useful, a method for determining suitable prior distributions for use in (14) is needed. In the joint BPC-PMC approach [1] we used Gaussian prior distributions with prior means set to the noisy speech mean found by PMC. The width of the prior was determined from the neighborhoods defined in [5]. We will now briefly review BPC-PMC before we introduce the new prior scaling method.

4.1. BPC with a Gaussian Prior

Let us first consider the case of a Gaussian prior with mean \( \alpha_m \) and variance \( \nu_d^2 \) for HMM state \( i \), mixture component \( m \) and feature vector element \( d \). (Note that the variance is independent of state and mixture component number.) Then, we can calculate (14) as

\[ \tilde{p}_{\text{lin}}(y_d) = \prod_{d=1}^{D} N(y_d; \mu_m, \Sigma_{\text{lin}}) N(\mu_m; \alpha_m, \nu_d^2) d\mu_m \]

\[ = N(y_d; \alpha_m, \nu_d^2 + \nu_d^2). \]

Plugging this into (13) results in the following state density for HMM state \( i \):

\[ \tilde{p}_i(y) = \sum_{m=1}^{M} w_{im} \prod_{d=1}^{D} N(y_d; \alpha_m, \nu_d^2 + \nu_d^2). \]

4.2. Setting the Hyperparameters

In the joint BPC-PMC approach, the prior mean is set to the noisy speech mean estimated by PMC, i.e.,

\[ \alpha_m = \mu_m^{\text{lin}}. \]
where $\mu_{imd}^y$ is the result of (7). The variance is set as follows. In order to avoid having to deal with the placement of different cepstral coefficients and their temporal derivatives in the feature vector, we define $q(d)$ to be a function that maps feature vector element $d$ to the order of the corresponding cepstral coefficient. The prior variance is set to

$$
\nu_d^2 = \begin{cases} 
\frac{1}{2}C^2 q(d)^{-2} p_{2q(d)} & \text{if } q(d) > 0 \\
\kappa_d \sigma_d & \text{if } q(d) = 0,
\end{cases}
$$

(18)

where $C > 0$ and $p \in (0, 1]$ are constants. The variance $\frac{1}{2}C^2 q(d)^{-2} p_{2q(d)}$ is a result of minimizing the Kullback-Leibler divergence between our Gaussian and a uniform distribution of width equal to $2Cq(d)^{-1}p_{2q(d)}$ [6]. For the zeroth order cepstral coefficient, the prior variance has been set to a constant $\kappa_d$ multiplied by the global standard deviation $\sigma_d$ from the clean speech training set for the corresponding coefficient (static, delta, or delta-delta).

### 4.3. Introducing the Scale Factor

As can be seen from (16), the variance is increased compared to the original state density. While this results in improved performance at low SNRs for most noise types, it causes a lot of insertion errors in the case of babble noise.

In order to reduce the number of insertion errors, we would like to reduce the variance during periods where there is no speech activity. Thus, we introduce a frame-dependent scale factor $\xi_t \in [0, 1]$ on the prior variance. The resulting prior for frame $t$ is then

$$p_t(\mu_{imd}; \phi_{imd}^t) = N(\mu_{im\bar{d}}; \alpha_{imd}, \xi_t \nu_d^2).$$

(19)

This prior results in the following state density for frame $t$ and HMM state $i$:

$$\hat{p}_{t,i}(y_t) = \sum_{m=1}^M w_m \prod_{d=1}^D N(y_{td} \delta_d; \mu_{imd} \xi, \sigma_{imd}^2 + \xi_t \nu_d^2).$$

(20)

The scale factor will be set to reflect our confidence that frame $t$ does contain speech. A high confidence will result in a scale factor with value close to 1, while a low confidence will give a value close to 0.

### 4.4. Setting the Scale Factor

In order to determine whether a given frame $t$ is likely to contain speech or not, we make use of the log-likelihood ratio (LLR)

$$LLR(y) = \log \frac{p(y | \Omega_{1})}{p(y | \Omega_{0})} = \log p(y | \Omega_{1}) - \log p(y | \Omega_{0}),$$

(21)

where the classes $\Omega_{0}$ and $\Omega_{1}$ represent speech present and speech absent respectively.

It has been shown [7] that for detecting speech it can be beneficial to use a likelihood ratio that takes into account several consecutive observations. More specifically, for a frame $t$, this is done by summing up the log likelihood ratios for the current frame and for $l$ frames in both directions. Thus, for frame $t$ we evaluate the sum of LLRs on the set $Y_{t-l+1}^{t+l} = \{y_{t-l}, ..., y_{t}, ..., y_{t+l} \}$ and get the following

$$LLR_t(Y_{t-l+1}^{t+l}) = \sum_{k=t-l}^{t+l} \log \frac{p(y_k | \Omega_{1})}{p(y_k | \Omega_{0})}$$

(22)

In order to calculate (22), we use the single Gaussian noise model estimated from the first $N$ frames of each file to model $\Omega_{0}$, while $\Omega_{l}$ is modeled by a Gaussian mixture model (GMM) for noisy speech. The parameters of this GMM are found by applying a first-order vector Taylor series (VTS) expansion [8] on a pre-trained clean speech GMM. By using VTS one can linearize the non-linear relationship between clean speech, noise, and noisy speech in the cepstral or log-spectral domain, and obtain closed-form expressions for approximations of the mean and covariance matrix of noisy speech. See e.g. [8] for more details. Then, using these models, and defining $\eta(t) = LLR_t(Y_{t-l+1}^{t+l})$, we can write the result as

$$\eta(t) = \sum_{k=t-l}^{t+l} \log \frac{\sum_{m=1}^M w_m N(y_k; \mu_{y,m}; \Sigma_{y,m})}{N(y_k; \mu_y, \Sigma_y)}.\quad (23)$$

The problem that remains is how to set the scale factor as a function of $\eta(t)$. The most straightforward approach is to perform voice activity detection by simply setting a threshold $\tau$ on the value of $\eta(t)$, i.e.,

$$\xi_t = \begin{cases} 1 & \text{if } \eta(t) > \tau \\
0 & \text{if } \eta(t) \leq \tau.\quad (24)$$

However, this means that we make a hard decision at each frame. An alternative solution is to use a continuous function $f : [0, 1] \rightarrow [0, 1]$ for mapping the LLR-values to scale factors, i.e., $\xi_t = f(\eta(t))$. A natural choice for $f$ is a sigmoid function. The parameters of the sigmoid have to be optimized in order to find a suitable slope and shift. In this paper we use a sigmoid on the form

$$\xi_t = \frac{1}{1 + \exp((-\frac{\eta(t) - \psi}{\tau}))},$$

(25)

where the parameter $\psi$ determines the shift and plays a similar role as the threshold in (24).

### 4.5. Summary of the Procedure

We will now sum up the steps needed to implement the proposed procedure. The following steps are performed for each file to be recognized:

1. Estimate noise parameters from the first $N$ frames
2. Using PMC as described Section 2, find estimates $\{\mu_{ymd}^n\}$ of the noisy speech mean for all mixture components $m$ in all HMM states $i$
3. For each frame $t$:
   a. Calculate the LLR value $\eta(t)$ using (23)
   b. Calculate the scale $\xi_t$ using either (24) or (25)
   c. For each HMM state $i$, mixture component $m$, and feature vector element $d$, set the parameters of the predictive distribution as follows:
      i. The mean is set to $\mu_{imd} = \mu_{ymd}^n$
      ii. The variance is set to $\sigma_{imd}^2 + \xi_t \nu_d^2$, with $\nu_d^2$ given in (18)

---

2Note for example, a $d$ that corresponds to the delta coefficient of cepstral coefficient 5 will be mapped to 5.
5. Experiments and Results

Experiments are performed on set A of the Aurora 2 database [9], which consists of US-English digits in presence of four different noise types: subway, babble, car, and exhibition. Each of these noise types are added at different SNRs, from 20 dB down to -5 dB in steps of 5 dB.

The HTK front-end was used for feature extraction. Standard Aurora 2 scripts were used for training the recognizer, with the exception that the zeroth order cepstral coefficient was used instead of log-energy. The feature vectors are 39-dimensional, consisting of static, delta, and delta-delta parameters. Note that for the calculation of the LLR in (23), only the static parameters were used. The average word accuracy for SNRs from 0 dB to 20 dB of the baseline system without model compensation was 56.19%.

The first 10 frames of each utterance were used for estimating the noise model used by BMC. Moreover, BMC was only applied to the means of static and delta parameters, while the means of the delta-delta parameters were left unchanged.

For the variance of the BMC prior, we used values of $C = 2$ and $\rho = 0.95$ for the constants in the first case of (18). The constants $\kappa_d$ were set to 1.0, 1.5, and 2.0 for the static, delta, and delta-delta parameters of the zeroth cepstral coefficient respectively.

The results for four different methods are shown in Table 1. The approach using hard decision prior scaling in (24) has been given the label “bpc-vad”, and the approach using the sigmoid in (25) has been given the label “bpc-sig”. For these approaches, the parameters $\tau$ and $\psi$ have been optimized to give the best average performance. The result was a $\tau = 26$ for “bpc-vad”, and a $\psi = 21$ for “bpc-sig”. We have included the average for -5 dB to 20 dB, in addition to the average for 0 dB to 20 dB which is the most commonly used.

First, comparing “pmc” and “bpc-pmc”, we can see that the latter is better for low SNRs at all noise types except babble. However, the performance for babble is so poor that the average word accuracy is lower than for “pmc”. When we look at the two approaches using prior scaling, we can see that the performance for babble is significantly increased compared to “bpc-pmc”. At the same time, the performance on subway and car is reduced. On exhibition we can observe an improvement (except at -5 dB). This is because exhibition noise also contains some background speech, and we are able to remove insertion errors by using prior scaling. On average we are able to improve the performance compared to “pmc”. The performance of “bpc-sig” is almost the same as for “bpc-vad”.

6. Conclusion

In this paper we improved on the previously presented joint BPC-PMC approach by introducing prior scaling. A frame-dependent scale factor was applied to the prior variance. The value of the scale factor reflects the confidence that the current frame contains speech. Prior scaling is effective in reducing the number of insertion errors when background speech is present.

The proposed approach gave a higher average word accuracy than both PMC and BPC-PMC.

7. References


Table 1: Word accuracies on set A of Aurora 2

<table>
<thead>
<tr>
<th>Noise</th>
<th>SNR</th>
<th>pmc</th>
<th>bpc-pmc</th>
<th>bpc-vad</th>
<th>bpc-sig</th>
</tr>
</thead>
<tbody>
<tr>
<td>subway</td>
<td>20</td>
<td>97.88</td>
<td>97.39</td>
<td>97.39</td>
<td>97.39</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>96.53</td>
<td>96.01</td>
<td>95.95</td>
<td>95.89</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>92.66</td>
<td>92.20</td>
<td>91.86</td>
<td>91.93</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>80.35</td>
<td>82.13</td>
<td>81.58</td>
<td>81.67</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>46.67</td>
<td>56.00</td>
<td>52.72</td>
<td>52.59</td>
</tr>
<tr>
<td></td>
<td>-5</td>
<td>21.31</td>
<td>25.88</td>
<td>23.24</td>
<td>23.30</td>
</tr>
<tr>
<td>babble</td>
<td>20</td>
<td>96.83</td>
<td>87.48</td>
<td>94.50</td>
<td>94.50</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>92.65</td>
<td>79.20</td>
<td>89.60</td>
<td>89.66</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>85.04</td>
<td>63.27</td>
<td>79.50</td>
<td>79.56</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>67.05</td>
<td>47.22</td>
<td>64.33</td>
<td>64.06</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>39.57</td>
<td>24.12</td>
<td>38.15</td>
<td>38.12</td>
</tr>
<tr>
<td></td>
<td>-5</td>
<td>17.11</td>
<td>8.16</td>
<td>16.60</td>
<td>16.48</td>
</tr>
<tr>
<td>car</td>
<td>20</td>
<td>98.30</td>
<td>97.64</td>
<td>97.55</td>
<td>97.52</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>96.96</td>
<td>96.87</td>
<td>96.75</td>
<td>96.72</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>91.50</td>
<td>92.48</td>
<td>91.92</td>
<td>91.92</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>70.27</td>
<td>76.74</td>
<td>74.80</td>
<td>74.56</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>31.05</td>
<td>45.75</td>
<td>38.56</td>
<td>38.41</td>
</tr>
<tr>
<td></td>
<td>-5</td>
<td>15.72</td>
<td>18.25</td>
<td>16.19</td>
<td>16.10</td>
</tr>
<tr>
<td>exhib.</td>
<td>20</td>
<td>97.41</td>
<td>96.36</td>
<td>96.82</td>
<td>96.85</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>95.83</td>
<td>94.35</td>
<td>95.09</td>
<td>95.09</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>91.05</td>
<td>89.02</td>
<td>89.85</td>
<td>89.82</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>75.75</td>
<td>75.13</td>
<td>77.45</td>
<td>77.23</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>42.27</td>
<td>46.74</td>
<td>47.39</td>
<td>47.42</td>
</tr>
<tr>
<td></td>
<td>-5</td>
<td>15.46</td>
<td>19.72</td>
<td>19.13</td>
<td>19.01</td>
</tr>
<tr>
<td>Avg. 0 – 20dB</td>
<td>79.28</td>
<td>76.81</td>
<td>79.59</td>
<td>79.55</td>
<td></td>
</tr>
<tr>
<td>Avg. -5 – 20dB</td>
<td>68.97</td>
<td>67.00</td>
<td>69.45</td>
<td>69.41</td>
<td></td>
</tr>
</tbody>
</table>


