PROSPECT Features and their Application to Missing Data Techniques for Robust Speech Recognition

Hugo Van hamme

K.U. Leuven – Dept. ESAT
Kasteelpark Arenberg 10, B-3001 Leuven, Belgium
hugo.vanhamme@esat.kuleuven.ac.be

Abstract

Missing data theory has been applied to the problem of speech recognition in adverse environments. The resulting systems require acoustic models that are expressed in the spectral domain, which leads to loss of accuracy. Cepstral Missing Data Techniques (CMDT) surmount this disadvantage, but require significantly more computation. In this paper, we study alternatives to the cepstral representation that lead to more efficient MDT systems. The proposed solution, PROSPECT features (Projected Spectra), can be interpreted as a novel speech representation, or as an approximation of the inverse covariance (precision) matrix of the Gaussian distributions modeling the log-spectra.

1. Introduction

Missing data techniques (MDT) for noise-robust speech recognition rely on the property that some regions in the spectrogram will be dominated by the speech signal that is to be recognized, while other regions will be dominated by unwanted signals such as noise or competing speech. A spectral mask defines where on the one hand spectral information is reliable and can be used as such, and on the other hand where it is unreliable and where an acoustic model trained on clean speech needs to be modified to assure a good match with the corrupted data. In [1], several methods for adapting continuous-density HMMs were presented.

A major drawback of missing data techniques is that the acoustic model must be expressed in the spectral domain since this is also the representation in which the masks have a simple formulation. While the cepstral domain is often used for reasons of accuracy, MDT recognizes work in with log-spectral features, which results in a loss in accuracy. Recently, research has begun to express the MDT-based acoustic models in the cepstral domain ([2], [3] and [4]) or even for arbitrary linear transforms of log-spectra [3]. While these cepstral MDT (CMDT) systems show superior accuracy on clean speech as well as superior robustness relative to their spectral predecessors, they require significantly more computation. In this paper, we present alternative MDT formulations through the introduction of the PROSPECT features such that the computational requirements of the CMDT systems are reduced while maintaining their accuracy.

This paper is organized as follows. Section 2 defines the PROSPECT representation of speech. Missing data techniques are first reviewed in section 3 and applied to the new representation. Methods for solving the resulting non-negative least squares problem are discussed in section 4. The resulting system is benchmarked on the AURORA-2 task in section 5.

2. PROSPECT: PROjected SPECTral features

The speech recognizer is assumed to have a mainstream HMM-based architecture with Gaussian mixture acoustic models. In the front-end, a low-resolution MEL-spectral representation is computed by a filter bank with D channels through windowing, framing, FFT and filter bank integration. At frame f, the output of the filter bank with center frequency f will be denoted by \( \mathbf{I}(f) \) for clean speech. The clean log-MEL-spectral features \( \mathbf{s} \) are then obtained by stacking \( \log(\mathbf{I}(f)) \) for all filter banks in a vector.

Log-spectral features are hard to model with a Gaussian Mixture Model (GMM) with diagonal covariance. The success of cepstral features for speech recognition is due to the property that the Discrete Cosine Transform (DCT) decorrelates the log-spectra. For reasons that will become clear in section 4, we are interested in a representation that uses only a low cepstral order. We consider the K cepstral features (\( c_0, \ldots, c_K \))

\[
\mathbf{c} = \mathbf{C}_K \mathbf{s}
\]

where \( \mathbf{C}_K \) is the K-by-D orthonormal DCT matrix. The residual spectrum \( \mathbf{d} \) is then (prime denoting matrix transpose and \( \mathbf{I}_D \) denoting the D-by-D identity matrix)

\[
\mathbf{d} = \mathbf{s} - \mathbf{C}_K \mathbf{c} = (\mathbf{I}_D - \mathbf{C}_K \mathbf{C}_K^T) \mathbf{s} = \mathbf{P}_d \mathbf{s}
\]

Hence, \( \mathbf{d} \) is the projection of \( \mathbf{s} \) onto the space perpendicular to the rows of \( \mathbf{C}_K \). The PROSPECT features are now defined as

\[
\mathbf{p} = \begin{bmatrix} \mathbf{c} \\ \mathbf{d} \end{bmatrix} = \begin{bmatrix} \mathbf{C}_K & \mathbf{I}_D - \mathbf{C}_K \mathbf{C}_K^T \\ \mathbf{0}_D & \mathbf{P}_d \end{bmatrix} \begin{bmatrix} \mathbf{s} \\ \mathbf{0}_K \end{bmatrix}
\]

We examine if these features can be modeled well by a GMM with diagonal covariance, even for small \( K \). Hence the likelihood of the \( t \)-th mixture component of HMM state \( q \) has the expression

\[
f(\mathbf{p} \mid i, q) = N(\mathbf{c} \mid i, q) N(\mathbf{d} \mid i, q)^\alpha
\]

where \( \alpha \) is a stream exponent and with (dropping the indices \( i \) and \( q \) in means and variances for notational convenience):

\[
N(\mathbf{c} \mid i, q) = \frac{1}{\sqrt{2\pi}} \prod_{j=0}^{K} \sigma_{c,j}^{-1} \exp\left(-\frac{1}{2} \sum_{j=0}^{K} (c_j - \mu_{c,j})^2 / \sigma_{c,j}^2 \right)
\]

and

\[
N(\mathbf{d} \mid i, q) = \frac{1}{\sqrt{2\pi}} \prod_{j=0}^{D} \sigma_{d,j}^{-1} \exp\left(-\frac{1}{2} \sum_{j=0}^{D} (d_j - \mu_{d,j})^2 / \sigma_{d,j}^2 \right)
\]

If all \( \sigma_{d,j} \) (\( j = 1 \ldots D \)) are estimated independently, we refer to a full acoustic model. If they are all pooled to the value

\[
\sigma_{d,0}^2 = \frac{1}{D} \sum_{j=1}^{D} \sigma_{d,j}^2
\]
we talk about a Euclidean or pooled acoustic model.

To suggest a value for $\alpha$, consider the conventional cepstral model which is obtained by replacing $K$ by a larger number $N$. Relative to a GMM in the cepstral domain, we replace

$$
\frac{1}{\sqrt{2\pi}^{N-NK\sum_j \sigma^2_{i,j}}} \exp \left( -\frac{1}{2} \sum_{j=1}^{K} \left( c_j - \mu_{i,j} \right)^2 \sigma^2_{i,j} \right)
$$

with (6). The latter contains a sum of $D$ normalized random variables in the exponent instead of $N-K$ in (8). This leads to a larger dynamic range of the state likelihood, which is compensated by a stream exponent $\alpha = (N-K)/D < 1$. In our experiments, $N=13$, $K=3$ and $D=22$, so $\alpha$ is fixed to 0.5.

We now verify experimentally how well PROSPECT features can be modeled by (4) on the 16kHz close-talk AURORA-4 large vocabulary benchmark using the clean data training set. Refer to [5] for more details on the recognizer, acoustic model training and experimental conditions. Table 1 shows the word error rate (WER) observed using a recognizer without MDT. A first baseline (rows “cep”) uses cepstral features and their velocity and acceleration with $N=13$. In a second baseline, the 13 static cepstra are substituted for 22 MEL filter bank log-power outputs (rows “MEL”). In both cases, the acoustic models are trained from scratch (rows “retrain”) or by single pass retraining (rows “1-pass”). In the latter case, the HMM model parameters are estimated by forced alignment using MIDA features as described in [5] while measuring the mean and diagonal covariance of each Gaussian on the parallel feature stream. Table 1 testifies that “1-pass” models perform better and will hence be used to generate the acoustic models for the PROSPECT features (3) as well. Notice that the MEL models lose in accuracy on clean data as well as in robustness. Since classical MDT systems described by (11) below would use these models we find a motivating argument to optimize the cepstral MDT techniques. Table 1 further compares the WER of pooled and full PROSPECT features for different values of $K$. A value as low as 2 or 3 yields excellent results compared to the cepstral models. Hence, the strongest correlations in a speech spectrum are due to the global spectral trend represented in the lower cepstral coefficients. The remaining covariance is well modeled by a diagonal structure of the residual spectrum $d$. The removal of the low order cepstrum from $d$ through projection is crucial since plain MEL log-spectra, which could be seen as PROSPECT with $K=0$, are poorly modeled by a diagonal covariance structure. Furthermore, the pooled model seems to be as accurate ($K > 2$) as the cepstral one but requires only $K+1$ variance parameters instead of $N$.

3. Missing Data Techniques

If the speech is corrupted by noise, we will denote the filter bank output by $Y_{i}(f)$ and the corresponding features by $y$. Similarly, if only the noise was processed by the front-end, the filter bank output is denoted by $N_{i}(f)$ and the feature vector by $n$. In missing data theory, it is now observed that

$$
y = \max(x, n)
$$

(9)

where the max-operator works element-wise. A missing data recognizer disparates a module that indicates for all $t$ which of the components of $y$ are reliable ($IS_{i}(f) \geq 2\text{SNR}_{i}(f)$) or unreliable ($IS_{i}(f) < \text{SNR}_{i}(f)$). The subscripts $u$ will be used to denote the unreliable elements of a vector. The $U$ unreliable speech components are estimated subject to the constraints derived from (9):

$$
s_u \leq y_s
$$

(10)

If $s$ is modeled by GMM with diagonal covariance in the log-spectral domain, then the Maximum Likelihood Estimate (MLE) of its unreliable components is given by

$$
\hat{s}_u = \min(y_s, \mu_u)
$$

(11)

where $\mu_u$ is the Gaussian mean in the log-spectral domain.

Solution (11) will be called the Spectral MDT (SMTP) solution. Though simple to compute, its main disadvantage is the loss in accuracy due to diagonal covariance matrix of spectral features. Indeed, when all features are reliable, the method reduces to evaluation of diagonal Gaussians in the spectral domain (row “MEL” in Table 1).

3.1. Cepstral Missing Data Technique (CMDT)

In [3] it was shown that this loss in accuracy is overcome by expressing the MDT problem as an MLE in the cepstral domain. For each Gaussian mixture component, this requires maximizing the log-likelihood over the $U$ components of $s_u$,

$$
\frac{1}{2} (s - \mu_u) \Lambda (s - \mu_u) + \text{SNR}_u \log \text{det} \Lambda_u
$$

(12)

with $\Lambda = C_u \Sigma_u C_u$ and $\Sigma_u$, the diagonal cepstral covariance. Since $N \leq D$, $\Lambda$ is rank-deficient and problem (12) is typically underdetermined. Hence it is regularized with a non-critical regularization constant $\lambda$:

$$
\Lambda = C_u \Sigma_u C_u + \lambda \Sigma_u^{-1}
$$

(13)

where $\Sigma_u$ is the diagonal spectral covariance matrix of the Gaussian. With the substitution $x = y - s$, the minimization of (12) is equivalent to minimization over $x_u$ of

$$
\frac{1}{2} x^2 \Lambda_u x_u - 2\Lambda_u y_u
$$

(14)
with
\[ 2b = C_{\alpha \beta}^{-1} (C_{\gamma \beta} (C_{\gamma \beta} - \mu_{\gamma}) + \lambda \Sigma_{\alpha \beta}^{-1} (y_{\alpha} - \mu_{\alpha}) ) \] (15)
and \( A_{0} (\Sigma_{0}) \) is obtained from \( A (\Sigma_{0}) \) by deleting all rows and columns corresponding to reliable features. Finally, the likelihood of the Gaussian is evaluated at the minimizer \( \mathbf{s} \) of (12):
\[
  f (\mathbf{s}, i, q) = \frac{\exp \left( \frac{1}{2} (\mathbf{s} - \mu_{\alpha})^{T} C_{\alpha}^{-1} (\mathbf{s} - \mu_{\alpha}) \right)}{\sqrt{2\pi}^{n} |\Sigma_{\alpha}|^{1/2}}
\] (16)

Although CMDT can be applied to static and dynamic cepstra jointly [3], for the sake of computational simplicity and limited accuracy gain, CMDT is applied to the static cepstra only in this paper.

3.2. MDT with PROSPECT features

When substituting (3) in (4), the log-likelihood becomes (within constants):
\[
  \frac{1}{2} (\mathbf{s} - \mu_{\alpha})^{T} C_{\alpha}^{-1} C_{\alpha} + \alpha \mathbf{P}_{\alpha}^{T} \Sigma_{\alpha}^{-1} \mathbf{P}_{\alpha} (\mathbf{s} - \mu_{\alpha})
\] (17)

Considered as a function of \( \mathbf{s} \), (17) also determines a Gaussian log-likelihood function where the matrix between brackets is an inverse covariance or precision matrix. Hence, the PROSPECT model defines an approximate form of the precision matrix of the spectral features and the CMDT optimization problem (12) can be formulated for it. The precision matrix \( \mathbf{A} \) is now of full rank and a regularization like (13) is not required. For the pooled acoustic model (7) and given that \( \mathbf{P}_{\alpha}^{T} \) is idempotent, (17) becomes:
\[
  \frac{1}{2} (\mathbf{s} - \mu_{\alpha})^{T} C_{\alpha}^{-1} C_{\alpha} + \alpha \mathbf{P}_{\alpha}^{T} \Sigma_{\alpha}^{-1} \mathbf{P}_{\alpha} (\mathbf{s} - \mu_{\alpha})
\] (18)

4. Solving the Non-negative Least Squares (NNLSQ) problem

Non-negative least squares problems are usually solved using “primal active set methods” [6], which require the unconstrained minimization of a sequence of quadratic functions with at most \( U \) unknowns (\( U \leq D \)). However, the matrix inversions involved in this optimization would require too much computation. Two alternative methods are presented, in section 4.1 and 4.2, and their convergence properties are examined in section 4.3.

4.1. Gradient descent

Matrix inversion or alternatively solving a linear set of equations can be omitted by resorting to gradient descent at the expense of potentially a slower convergence. The search is started from the SMDT solution (11). The search direction \( \mathbf{g} \) is derived from the cost gradient \( \mathbf{g} \) by zeroing those components that would violate the constraint, i.e. if the gradient component is negative and the corresponding speech estimate is on the constraint boundary (12). The expression for the gradient of the cost function (12) at \( \mathbf{s} \) is:
\[
  \mathbf{g} = \mathbf{A} (\mathbf{s} - \mu_{\alpha})
\] (19)

Making a step then entails replacing \( \mathbf{s} \) by its new estimate:
\[
  \mathbf{s} \leftarrow \mathbf{s} - \beta \mathbf{g}
\] (20)

The optimal step size
\[
  \beta = \frac{\mathbf{g}^{T} \mathbf{g}}{\mathbf{g}^{T} \mathbf{A} \mathbf{g}}
\] (21)

is reduced to \( \bar{\beta} \) such that all components of \( -\bar{\beta} \mathbf{g} \) remain smaller than the maximum step size \( \mathbf{y} \) that would violate (10). Each step then reduces the cost by
\[
  \bar{\beta} \left( \frac{\mathbf{g}^{T} \mathbf{g}}{2 \mathbf{g}^{T} \mathbf{A} \mathbf{g}} \right)
\] (22)

The computational complexity of a single gradient step for the three different choices of the precision matrix is listed in Table 2 as the number of multiply-accumulate operations. Cost function, gradient and step size can be computed from the listed items in a negligible \( O(K) \), \( O(U) \) or \( O(D) \) operations. An exception is formed if no iteration is performed, i.e. when the Gaussian is evaluated at the SMDT solution. By exploiting the structure of the precision matrix, additional computational gains can be obtained. The resulting computational cost is listed in the row “No iteration” of Table 2. An example with typical values \( N=13 \), \( K=3 \), \( D=23 \) and \( U=16 \) (an average measured on data) is given for the totals. We observe that PROSPECT features always reduce the effort, especially with variance pooling.

4.2. Multiplicative updates

In [7], a method of surprising simplicity and elegance for solving the non-negative least squares problem (14) is described. Let \( A_{\alpha}^{-1} (A_{\alpha}) \) be obtained from \( A_{\alpha} \) by setting all negative (positive) entries to zero. Then the update is given by:
\[
  \mathbf{x}_{\alpha} \leftarrow \mathbf{x}_{\alpha} \odot \left[ \mathbf{b} + \left[ (A_{\alpha}^{-1} A_{\alpha}) \odot (A_{\alpha}^{-1} A_{\alpha}) \right] \cdot (A_{\alpha}^{-1} A_{\alpha}) \right]
\] (23)

where \( \odot \) and \( / \) denote element-wise multiplication and division. The computational load of each iteration is dominated by two matrix multiplications or \( 2U^{2} = 512 \) in Table 2) operations and \( U \) square roots. A major drawback of the method is that we do not dispose of methods to factor \( A_{\alpha}^{+} \) and \( A_{\alpha}^{-1} \) and therefore they need to be precalculated and stored in full for all Gaussians.

Like in the gradient method, the initial point of the iteration is also derived from the SMDT solution, but since multiplicative updates cannot change a zero element in \( \mathbf{x}_{\alpha} \), we add a constant equivalent to 0.5dB to all zero elements. This allows the algorithm to move away from the constraint boundary if that decreases to cost function.

<table>
<thead>
<tr>
<th>Quantity</th>
<th>Cepstral Eq. (13)</th>
<th>Full PROSPECT eq. (17)</th>
<th>Pooled PROSPECT eq. (18)</th>
</tr>
</thead>
<tbody>
<tr>
<td>( C(s-\mu) ), ( C^{-1}(s-\mu) )</td>
<td>( NU )</td>
<td>( KU )</td>
<td>( KU )</td>
</tr>
<tr>
<td>( C^{-1}(s-\mu) )</td>
<td>( NU )</td>
<td>( KU )</td>
<td>( KU )</td>
</tr>
<tr>
<td>( 1-C^{-1}(s-\mu) )</td>
<td>( KD )</td>
<td>( KU )</td>
<td>( KU )</td>
</tr>
<tr>
<td>( P^{-1}(s-\mu) )</td>
<td>( K(U+D) )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total per iter. example</td>
<td>(2N+1)U</td>
<td>( K(3U+2D) )</td>
<td>( 3KU )</td>
</tr>
<tr>
<td></td>
<td>432</td>
<td>282</td>
<td>144</td>
</tr>
<tr>
<td>No iteration example</td>
<td>(N+1)U</td>
<td>( K(U+D) )</td>
<td>( 2KU )</td>
</tr>
<tr>
<td></td>
<td>208</td>
<td>117</td>
<td>96</td>
</tr>
<tr>
<td>Storage</td>
<td>( N+D )</td>
<td>( K+D )</td>
<td>( K+1 )</td>
</tr>
</tbody>
</table>

Table 2: approximate computational complexity of a single gradient iteration and storage per Gaussian required for various choices of the precision matrix.
4.3. Convergence properties

The convergence behavior of the gradient descent and multiplicative update algorithm are studied on the AURORA-2 test set A with train noise. The reader is referred to section 5 for more details on the experimental setup. For practical matters, the gradient method converges in 1 or 2 iterations (Table 3). Even for 0 iterations, a good level of robustness is obtained (see also [3], [2] and [4]) but the additional price for at least one iteration seems worth while.

For both the cepstral (Table 4) and full diagonal covariance PROSPECT (Table 5) formulation, it is clear that 5 iterations of the multiplicative update algorithm suffice to attain convergence and that as few as 2 iterations result in only a small loss in accuracy. It seems that gradient descent is the better choice for an efficient implementation, though the code for multiplicative updates is more elegant.

### Table 3: accuracy versus the number of gradient descent iterations for the train noise of set A of the AURORA 2 task using the full PROSPECT model (17) (K = 3).

<table>
<thead>
<tr>
<th>SNR (dB)</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>5</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>99.02</td>
<td>99.29</td>
<td>99.29</td>
<td>99.36</td>
<td>99.36</td>
</tr>
<tr>
<td>15</td>
<td>97.97</td>
<td>98.80</td>
<td>98.99</td>
<td>98.96</td>
<td>98.89</td>
</tr>
<tr>
<td>10</td>
<td>96.85</td>
<td>97.33</td>
<td>97.48</td>
<td>97.42</td>
<td>97.70</td>
</tr>
<tr>
<td>5</td>
<td>90.85</td>
<td>92.54</td>
<td>92.48</td>
<td>92.20</td>
<td>92.05</td>
</tr>
</tbody>
</table>

### Table 4: accuracy versus the number of multiplicative updates for the train noise of set A of the AURORA 2 task for cepstral precision matrix formulation (13) (N = 15).

<table>
<thead>
<tr>
<th>SNR (dB)</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>5</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>98.25</td>
<td>98.65</td>
<td>98.74</td>
<td>98.74</td>
<td>98.71</td>
</tr>
<tr>
<td>10</td>
<td>96.47</td>
<td>96.78</td>
<td>97.02</td>
<td>97.14</td>
<td>97.33</td>
</tr>
<tr>
<td>5</td>
<td>91.03</td>
<td>91.43</td>
<td>91.62</td>
<td>91.50</td>
<td>91.34</td>
</tr>
</tbody>
</table>

### Table 5: accuracy versus the number of multiplicative updates for the train noise of set A of the AURORA 2 task using the full PROSPECT model (17) (K = 3).

<table>
<thead>
<tr>
<th>SNR (dB)</th>
<th>full</th>
<th>pooled</th>
<th>full</th>
<th>pooled</th>
<th>full</th>
<th>pooled</th>
<th>full</th>
<th>Pooled</th>
</tr>
</thead>
<tbody>
<tr>
<td>K=1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>-15.0</td>
<td>-16.4</td>
<td>-3.4</td>
<td>-8.1</td>
<td>-7.8</td>
<td>-7.5</td>
<td>-7.4</td>
<td>-15.1</td>
</tr>
<tr>
<td>15</td>
<td>-11.9</td>
<td>-22.2</td>
<td>6.7</td>
<td>-2.7</td>
<td>0.9</td>
<td>-0.5</td>
<td>-3.2</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>-11.2</td>
<td>-18.0</td>
<td>6.0</td>
<td>-0.6</td>
<td>1.5</td>
<td>1.0</td>
<td>5.9</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>-1.0</td>
<td>8.9</td>
<td>6.1</td>
<td>1.5</td>
<td>8.6</td>
<td>8.0</td>
<td>4.1</td>
<td></td>
</tr>
</tbody>
</table>

### Table 6: relative WER improvement in % of PROSPECT features with full and pooled variance relative to the cepstral MDT baseline.

5. Robustness experiments

The robustness of the PROSPECT features with MDT is now examined on the AURORA-2 task. We use a 23 channel MEL filter bank identical to the AURORA WI-007 implementation and "oracle" missing data masks computed from the true speech and noise signals. The PROSPECT acoustic models are derived from the "advanced front end with complex back end" baseline through single-pass retraining. Further details on the experimental conditions are described in [3]. Table 6 lists the relative improvement in WER averaged over all 4 noise conditions of test set A of MDT-based recognition applied to projected features with full (17) and pooled variance (18) relative to the cepstral baseline (13). The analysis is limited to SNRs of 5dB and better because the error rates at lower SNR are so high they become of little relevance. The PROSPECT features use a single iteration of the fast gradient descent method, while the cepstral method uses 10 iterations with multiplicative updates. The projected features seem to offer a better robustness in combination with the CMDT technique. At high SNR, there is some loss in accuracy, especially with pooling, which was not observed on the large vocabulary AURORA-4 test. On the other hand, the robustness seems to improve over cepstra with PROSPECT features.

6. Conclusions

A novel representation of speech, PROSPECT, that is an alternative to the cepstra and that is computed as a linear transform of the log-spectra, was proposed. On the small vocabulary AURORA-2 recognition task, a small loss in accuracy was observed at high SNR, but the robustness was better. On a large vocabulary task, PROSPECT showed comparable accuracy as cepstra, even if the variances were pooled, leading to a reduction in the number of variance parameters to be estimated. PROSPECT features are particularly useful in Missing Data recognition systems, where the computational load can be reduced thanks to the low cepstral order.

7. References