

Semi-tied full deviation matrices for Laplacian density models

Christoph Neukirchen

Philips Research Laboratories, Weisshausstr. 2, 52066 Aachen, Germany

Christoph.Neukirchen@philips.com

Abstract

The Philips speech recognition system uses mixtures of Laplacian densities with diagonal deviations to model acoustic feature vectors. Such an approach neglects the correlations between different feature components that typically exist in the acoustic vectors. This paper extends the conventional Laplacian approach to model the between-feature interdependencies explicitly. These extensions either lead to a full deviation matrix model or to an integrated feature space transformation similar to the semi-tied covariances for Gaussian densities. Both methods can be efficiently implemented by exploiting a strong tying of the feature transformations and the deviation matrices, respectively. The novel approach is evaluated on two different digit string recognition tasks.

1. Introduction

Most speech recognition systems are based on HMMs with mixtures of Gaussian densities; the Philips system makes use of mixtures of Laplacian densities [1]. A more general density model which contains both kinds of densities as special cases makes use of the Minkowski- R metric and is given by (univariate case):

$$\log p(x) = - \left(\left| \frac{x - \mu}{\lambda} \right| \right)^R + K_R(\lambda) \quad (1)$$

x is the feature variable, μ is the center parameter and λ is the deviation parameter of the density. The normalization constant $K_R(\lambda)$ ensures that $p(x)$ integrates to unity. For the case $R=2$, Eq. 1 is a Gaussian density with $\lambda = \sqrt{2}\sigma$ describing the model covariance; for $R=1$, Eq. 1 equals the Laplacian density.

In speech recognition a large set of basic densities (which are indexed by m here) is used to model multi-dimensional acoustic feature vectors \mathbf{x} which might contain some degree of correlation across the D vector dimensions. A natural extension of Eq. 1 to a multi-dimensional density model is given by:

$$\log p_m(\mathbf{x}) = - \left(\left\| \Lambda^{(m)-1} \cdot (\mathbf{x} - \boldsymbol{\mu}^{(m)}) \right\|_R \right)^R + K_R(\Lambda^{(m)}) \quad (2)$$

here $\|\cdot\|_R$ denotes taking the vector R -norm, $\boldsymbol{\mu}^{(m)}$ is the $D \times 1$ -dimensional center vector and $\Lambda^{(m)}$ is a $D \times D$ -dimensional deviation matrix; in general both parameter structures are specific to the m -th density. The off-diagonal components in the matrix $\Lambda^{(m)}$ describe the interdependencies between different components in the feature vectors. In practice, the approach of Eq. 2 using full deviation matrices is not attractive for large systems because of limitations in speed, size and parameter estimation robustness. To avoid these problems, more constrained models are preferred that either enforce a diagonal structure for the deviation matrix (i.e. neglecting the between-feature interdependencies) or/and heavily tie the same deviation parameters across many densities [1].

A method that explicitly models some cross-feature correlations while keeping the advantages of diagonal covariance structures is the semi-tied covariance or MLLT model proposed for Gaussian densities in [2] and [3]. Application of a semi-tied deviation structure to Eq. 2 leads to the generalized model:

$$\log p_m(\mathbf{x}) = \frac{1}{2} \log \left((\det \mathbf{A}^{(c_m)})^2 \right) + K_R(\Lambda_{\text{diag}}^{(m)}) - \left(\left\| \Lambda_{\text{diag}}^{(m)-1} \cdot \mathbf{A}^{(c_m)} \cdot (\mathbf{x} - \boldsymbol{\mu}^{(m)}) \right\|_R \right)^R \quad (3)$$

The parameters which are specific to the m -th model are the center $\boldsymbol{\mu}^{(m)}$ and the diagonal deviation matrix $\Lambda_{\text{diag}}^{(m)}$. The set of basic models is grouped into several transformation classes; the class for the m -th model is denoted c_m . Each class c makes use of a $D \times D$ transformation matrix $\mathbf{A}^{(c)}$ which is common to all densities in this class. The semi-tied deviation model can be interpreted as an optimal transformation of the feature space ([3]) which leads to an efficient implementation of Eq. 3 by performing a class-wise multiplication of the feature vectors.

Finding direct solutions for the parameter estimates of the generalized models Eq. 2 and Eq. 3 is non-trivial in general. However, in the following sections a procedure is given that finds an approximate Maximum Likelihood solution for the parameters in the case of a Laplacian model with a semi-tied deviation as well as with a full deviation matrix. Finally, experimental results are presented for this novel model class in an efficient configuration.

2. Semi-tied full deviation model for Laplacian densities

For the case $R=1$, Eq. 3 equals a Laplacian density with semi-tied deviation model which can be decomposed into a sum over the transformed vector components:

$$\log p_m(\mathbf{x}) = \frac{1}{2} \log \left((\det \mathbf{A}^{(c_m)})^2 \right) + K_1(\Lambda_{\text{diag}}^{(m)}) - \sum_{d=1}^D \frac{1}{\lambda_d^{(m)}} \left| \mathbf{a}_d^{(c_m)} \cdot (\mathbf{x} - \boldsymbol{\mu}^{(m)}) \right| \quad (4)$$

with $\lambda_d^{(m)} > 0$ being the d -th diagonal component of $\Lambda_{\text{diag}}^{(m)}$, and $\mathbf{a}_d^{(c_m)}$ is the d -th row ($1 \times D$ dimensional) of the matrix $\mathbf{A}^{(c_m)}$.

2.1. Maximum Likelihood parameter estimation

In the speech recognition system the densities in Eq. 4 are used in a Laplacian mixture based HMM framework. The system parameters (here we focus on $\boldsymbol{\mu}^{(m)}$, $\Lambda_{\text{diag}}^{(m)}$ and $\mathbf{A}^{(c)}$ only) are trained in a Maximum Likelihood manner by an EM-algorithm. The M-step requires the optimization of the auxiliary function

with respect to the model (\mathcal{M}) parameters:

$$Q(\mathcal{M}, \hat{\mathcal{M}}) = \sum_m \sum_t \gamma_m(t) \log(p_m(\mathbf{x}(t))) \quad (5)$$

with $\gamma_m(t) = P(m|t, \hat{\mathcal{M}}, \mathcal{X})$ being the posterior probability of the m -th density at time t based on the old model $\hat{\mathcal{M}}$.

Since there is no simple direct optimization of Eq. 5, similar to [2] a double nested loop is applied which aims at increasing Eq. 5 locally by optimizing the matrices row-by-row iteratively:

1. init: set all transformations $\mathbf{A}^{(c)} = \mathbf{I}$ (identity matrix)
2. Estimate centers $\boldsymbol{\mu}^{(m)}$ and diagonal deviations $\Lambda_{\text{diag}}^{(m)}$ (and keep all transformation matrices $\mathbf{A}^{(c)}$ fixed)
3. Estimate transformation matrices $\mathbf{A}^{(c)}$ of all classes c (and keep centers $\boldsymbol{\mu}^{(m)}$ and diag. deviations $\Lambda_{\text{diag}}^{(m)}$ fixed)
 - (a) select a new matrix row index d
 - (b) optimize $\mathbf{a}_d^{(c)}$, i.e. d -th row of $\mathbf{A}^{(c)}$ (and keep all other rows $\mathbf{a}_i^{(c)}$ in matrix $\mathbf{A}^{(c)}$ fixed)
 - (c) Go to step 3a until convergence criterion satisfied
4. Go to step 2 until convergence criterion satisfied

The estimation of $\boldsymbol{\mu}^{(m)}$ and $\Lambda_{\text{diag}}^{(m)}$ in step 2 is performed by using the standard Baum-Welch estimates based on the transformed training vectors. To find proper values for the d -th row of $\mathbf{A}^{(c)}$ in step 3b the derivative of Eq. 5 with respect to the row $\mathbf{a}_d^{(c)}$ is set to zero.

2.2. Matrix row optimization

When differentiating the auxiliary function, the derivative of the semi-tied deviation Laplacian model (Eq. 4) is involved:

$$\nabla_{\mathbf{a}_d^{(c)}} (\log p_m(\mathbf{x})) = \frac{1}{\det \mathbf{A}^{(c)}} \mathbf{cof}_d(\mathbf{A}^{(c)}) - \frac{\text{sign}[\mathbf{a}_d^{(c)} \cdot (\mathbf{x} - \boldsymbol{\mu}^{(m)})]}{\lambda_d^{(m)}} (\mathbf{x} - \boldsymbol{\mu}^{(m)})^T \quad (6)$$

here $\mathbf{cof}_d(\mathbf{A}^{(c)})$ denotes the $1 \times D$ dimensional d -th row of cofactors of $\mathbf{A}^{(c)}$. Finally, the optimization of Eq. 5 with respect to the d -th matrix row $\mathbf{a}_d^{(c)}$ leads to the expression (using the equality $\det(\mathbf{A}^{(c)}) = \mathbf{a}_d^{(c)} \cdot (\mathbf{cof}_d(\mathbf{A}^{(c)}))^T$):

$$\frac{\beta^{(c)}}{\mathbf{a}_d^{(c)} \cdot (\mathbf{cof}_d(\mathbf{A}^{(c)}))^T} \mathbf{cof}_d(\mathbf{A}^{(c)}) = \mathbf{f}(\mathbf{a}_d^{(c)}) \quad (7)$$

with the count variable for the transformation class c :

$$\beta^{(c)} = \sum_{m \in c} \sum_t \gamma_m(t) \quad (8)$$

and $\mathbf{f}(\mathbf{a}_d^{(c)})$ as a $1 \times D$ dimensional vector valued function:

$$\mathbf{f}(\mathbf{a}_d^{(c)}) = \sum_{m \in c} \frac{1}{\lambda_d^{(m)}} \sum_t \gamma_m(t) \text{sign}[\mathbf{a}_d^{(c)} \cdot \mathbf{y}^{(m)}(t)] (\mathbf{y}^{(m)}(t))^T \quad (9)$$

$\mathbf{y}^{(m)}(t) = \mathbf{x}(t) - \boldsymbol{\mu}^{(m)}$ is the $D \times 1$ dimensional difference between the feature vector at time t and the m -th Laplacian center parameter vector.

In Eq. 6 and Eq. 9 the expression 'sign[...]' represents the sign of the scalar product of the vectors $\mathbf{a}_d^{(c)}$ and $\mathbf{y}^{(m)}(t)$ which equals the sign of the cosine of the angle between both vectors. Thus, $\mathbf{f}(\mathbf{a}_d^{(c)})$ is a step function with discontinuities on all hyperplanes which are orthogonal to those difference vectors $\mathbf{y}^{(m)}(t)$ which are involved in the sums of Eq. 9.

2.2.1. Approximating the step function

Due to the non-continuous characteristics of the function $\mathbf{f}(\mathbf{a}_d^{(c)})$ the existence of a solution to Eq. 7 is not guaranteed. In the cases a solution exists there is no simple closed form expression for the optimal row vector $\mathbf{a}_d^{(c)}$ because of the non-linear dependencies in $\mathbf{f}(\mathbf{a}_d^{(c)})$.

To enforce an approximate closed form solution to Eq. 7 the sign function in Eq. 9 is replaced by the quasi-linear approximation $s(\mathbf{a}_d^{(c)})$:

$$\text{sign}[\mathbf{a}_d^{(c)} \cdot \mathbf{y}^{(m)}(t)] \approx s(\mathbf{a}_d^{(c)}) = \frac{\mathbf{a}_d^{(c)}}{\|\mathbf{a}_d^{(c)}\|_2} \cdot \tilde{\mathbf{s}}^{(m,d)}(t) \quad (10)$$

The approximating function $s(\mathbf{a}_d^{(c)})$ and the original sign function share the property of being invariant in the length of the vector $\mathbf{a}_d^{(c)}$ (only the vector angle matters). The $D \times 1$ dimensional vector $\tilde{\mathbf{s}}^{(m,d)}(t)$ contains the parameters of the quasi-linear approximation. It is set such that $s(\mathbf{a}_d^{(c)})$ equals the first order Taylor series of a smoothed version of the original sign function at $\mathbf{a}_d^{(c)} = \mathbf{e}_d$ (with \mathbf{e}_d being the d -th $1 \times D$ dimensional unit vector, which is the starting point for $\mathbf{a}_d^{(c)}$ at step 3 in the iterative scheme of Sec. 2.1).

The smoothed step function to be fitted is constructed from the hyperbolic tangent sigmoid function:

$$\tanh\left(\alpha \frac{\mathbf{a}_d^{(c)}}{\|\mathbf{a}_d^{(c)}\|_2} \cdot \frac{\mathbf{y}^{(m)}(t)}{\|\mathbf{y}^{(m)}(t)\|_2}\right) \quad (11)$$

This sigmoid function's arguments are chosen such that Eq. 11 does neither depend on the scaling of the feature vector differences $\mathbf{y}^{(m)}(t)$ nor on the scaling of the matrix rows $\mathbf{a}_d^{(c)}$. The constant α determines the smoothness of Eq. 11 and it is set quite large (in the range 10^2 – 10^3) to make Eq. 11 follow the sign function reasonable well; for the iterative matrix row optimization the exact value for α was not critical.

The first order fit of $s(\mathbf{a}_d^{(c)})$ to Eq. 11 results in setting the i -th component of the parameter vector $\tilde{\mathbf{s}}^{(m,d)}(t)$ to:

$$\tilde{\mathbf{s}}_i^{(m,d)}(t) = \begin{cases} \tanh\left(\frac{\alpha y_d^{(m)}(t)}{\|\mathbf{y}^{(m)}(t)\|_2}\right) & (i = d) \\ \alpha y_i^{(m)}(t) \left(1 - \tanh^2\left(\frac{\alpha y_d^{(m)}(t)}{\|\mathbf{y}^{(m)}(t)\|_2}\right)\right) & (i \neq d) \end{cases}$$

with $y_i^{(m)}(t)$ as the i -th component of the vector $\mathbf{y}^{(m)}(t)$.

After using the approximation Eq. 10 instead of the original sign function, Eq. 9 simplifies into a quasi-linear expression:

$$\mathbf{f}(\mathbf{a}_d^{(c)}) \approx \frac{\mathbf{a}_d^{(c)}}{\|\mathbf{a}_d^{(c)}\|_2} \cdot \tilde{\mathbf{F}}^{(c,d)} \quad (12)$$

which contains $\tilde{\mathbf{F}}^{(c,d)}$ as the d -th accumulator matrix for the transformation class c . All matrices $\tilde{\mathbf{F}}^{(c,d)}$ can be filled simultaneously by running over the training data

$$\tilde{\mathbf{F}}^{(c,d)} = \sum_{m \in c} \frac{1}{\lambda_d^{(m)}} \sum_t \gamma_m(t) \tilde{\mathbf{s}}^{(m,d)}(t) \cdot (\mathbf{y}^{(m)}(t))^T \quad (13)$$

In total, for each transformation class, D accumulator matrices of dimension $D \times D$ have to be stored. This equals the same number of accumulator elements to be stored as the memory efficient method for the estimation of Gaussian semi-tied covariances described in [2].

2.2.2. Setting the transformation matrix coefficients

When substituting the step function $\mathbf{f}(\mathbf{a}_d^{(c)})$ by the quasi-linear vector function in Eq. 12, expression Eq. 7 can be written as:

$$\beta^{(c)} \|\mathbf{a}_d^{(c)}\|_2 \mathbf{cof}_d(\mathbf{A}^{(c)}) \cdot \tilde{\mathbf{F}}^{(c,d)-1} = \mathbf{a}_d^{(c)} \cdot (\mathbf{cof}_d(\mathbf{A}^{(c)}))^T \mathbf{a}_d^{(c)} \quad (14)$$

The optimal d -th row vector $\mathbf{a}_d^{(c)}$ is given implicitly by Eq. 14. A solution to Eq. 14 is:

$$\mathbf{a}_d^{(c)} = \frac{\beta^{(c)} \|\mathbf{cof}_d(\mathbf{A}^{(c)}) \cdot \tilde{\mathbf{F}}^{(c,d)-1}\|_2 \mathbf{cof}_d(\mathbf{A}^{(c)}) \cdot \tilde{\mathbf{F}}^{(c,d)-1}}{\mathbf{cof}_d(\mathbf{A}^{(c)}) \cdot \tilde{\mathbf{F}}^{(c,d)-1} \cdot (\mathbf{cof}_d(\mathbf{A}^{(c)}))^T} \quad (15)$$

Thus, similar to the optimization in [2] the setting for the row $\mathbf{a}_d^{(c)}$ depends on all other rows in $\mathbf{A}^{(c)}$ via the cofactors. Therefore, after assigning a new row in $\mathbf{A}^{(c)}$ in each iteration step the vector $\mathbf{cof}_d(\mathbf{A}^{(c)})$ needs to be updated.

Due to the usage of the quasi-linear approximation the row estimate according to Eq. 15 does not guarantee to optimize the auxiliary function Eq. 5 in all cases. Sometimes, even a local decrease of likelihood can be observed. In these situations the matrices $\mathbf{A}^{(c)}$ typically tend to converge towards solutions with extremely large off-diagonal coefficients (which differs significantly from the identity matrices at the starting point). To prevent running into such likelihood decreasing solutions the estimate Eq. 15 is accepted as the new d -th row of $\mathbf{A}^{(c)}$ only when the largest element in $\mathbf{a}_d^{(c)}$ is at the d -th position, otherwise the d -th matrix row remains at its former value.

3. Full deviation matrix model for Laplacian densities

For Gaussian density based HMM systems with full covariance matrices a direct solution for estimating covariances is commonly used. For Laplacian based systems this situation corresponds to the special case of Eq. 2 with $R = 1$ but there is no such direct solution known for estimating the parameters of the full deviation matrices $\mathbf{A}^{(m)}$.

However, a similar procedure to the one described above for semi-tied deviation models can be used to estimate full deviation matrices for Laplacian densities in a row-by-row mode: the full deviation model Eq. 2 may be interpreted as a special case of the semi-tied structure Eq. 3 when keeping the diagonal elements in $\mathbf{\Lambda}_{\text{diag}}^{(m)}$ fixed to unity (i.e. $\mathbf{\Lambda}_{\text{diag}}^{(m)} = \mathbf{I}$) and identifying the inverse full deviation matrix $\mathbf{\Lambda}^{(m)-1}$ in Eq. 2 as a density specific feature space transformation matrix $\mathbf{A}^{(m)}$. Then the rows of this feature space transformation matrix $\mathbf{\Lambda}^{(m)-1}$ can be iteratively estimated by setting each row according to Eq. 15 by using the cofactors of the current estimate of $\mathbf{\Lambda}^{(m)-1}$ and using the density specific accumulator matrices

$$\tilde{\mathbf{F}}^{(m,d)} = \sum_t \gamma_m(t) \tilde{\mathbf{s}}^{(m,d)}(t) \cdot (\mathbf{y}^{(m)}(t))^T \quad (16)$$

for the m -th Laplacian model instead of the original transformation class specific accumulators. When full deviation matrices are heavily tied across several Laplacians (which is the only scenario that can be run-time efficiently implemented) it is sufficient to collect the accumulator matrices Eq. 16 for each deviation matrix to estimate by summing over all Laplacians involved.

4. Experimental results

The recognition performance on two different speaker-independent digit string recognition tasks is evaluated to compare Laplacian based mixture density HMMs with full and semi-tied deviations against diagonal deviations: i) the male portion of the Sietill task, ii) the Aurora2 task.

The systems' feature preprocessing extracts 12 MFCCs every 16ms. For the Sietill system the first 8 delta features are appended (i.e. $D = 20$). For the Aurora2 system 12 deltas are used (i.e. $D = 24$) and to cope with the noisy environment, a non-linear spectral subtraction (NSS) and a SNR-normalization are applied first (see [5]). To optionally improve system performance, LDA (see [4]) can be applied to supervectors which are spliced together from the original MFCC vector plus a feature vector from the left and from the right frame. The LDA classes correspond to the HMM states which are generated from a Viterbi alignment using the original MFCC systems.

Whole word HMMs are constructed for digit modeling. Parameters are estimated in a Maximum Likelihood manner using the Viterbi approximation. No kind of adaptation method (neither to environment nor to speakers) is used in the experiments. Two kinds of systems are constructed for both tasks: a small system with 4 Laplacian densities per mixture and a larger system with approx. 30 Laplacian densities per mixture on average.

In the experiments, diagonal deviation matrix models (with different degrees of tying) are compared to semi-tied and full deviation models. To allow an efficient implementation in a single matrix feature-space-based transformation framework, the systems reported here make either use of a single transformation matrix (i.e. one transformation class) for the semi-tied models or of one globally tied full deviation matrix model. For simplicity, these models are initialized from fully trained diagonal deviation systems. Then, four iteration steps of the outer loop from Sec. 2.1 (which also involves reestimation of the Laplacian center parameters and the diagonal deviation parameters of the semi-tied system) are run. Hence, in contrast to the preferred method in [2], in these experiments the transformation matrix estimation is not embedded into the density splitting process.

4.1. Sietill male: german digit strings over telephone lines

The male portion of the Sietill database contains 22631 digits in 6886 utterances for training and 22881 digits in 6938 utterances for testing. The recognition system consists of 256 mixtures of either 1k Laplacians in total for the small system (Tab. 1) or 8k Laplacians in total for the large system (Tab. 2). To compare the influence of feature transformations the LDA-transformed vectors are projected to 20 dimensions which equals the original MFCC vector size. The diagonal deviation matrices are either density specific (untied) or globally tied; the full covariance matrix is globally tied and the semi-tied models make use of density specific diagonal deviations and of a single full transformation matrix.

In all cases (small and large systems) the LDA transformation of the MFCC features leads to a 10% to 15% relative WER reduction. The performance gains in the small (1k) systems when going from a single diagonal deviation matrix to a single globally tied full matrix is better than using 1024 different density specific diagonal deviations. For the large (8k) systems, the corresponding gains are equal which might be caused by the larger increase of the number of parameters for the density specific diagonal deviation systems. In all systems the best performance is achieved in the semi-tied configuration with a single full transformation matrix and 1024 or 8086 diagonal de-

vations, respectively. However, the relative improvement of the semi-tied configuration versus the globally tied diagonal baseline setup is smaller for the 8k-systems compared to the 1k-systems; this might be caused by the stronger implicit modeling of feature correlation when using mixtures with 30 Laplacian components on average in the large system.

Table 1: WER of small Sietill system, 1024 Laplacian densities

| deviation matrices | | | features | |
|--------------------|----------|------------|----------|---------|
| # | tying | matr. type | MFCC(20) | LDA(20) |
| 1 | global | diagonal | 3.05 | 2.72 |
| 1024 | no tying | diagonal | 3.05 | 2.64 |
| 1 | global | full | 2.92 | 2.46 |
| 1024 | | semi-tied | 2.67 | 2.25 |

Table 2: WER of large Sietill system, 8086 Laplacian densities

| deviation matrices | | | features | |
|--------------------|----------|------------|----------|---------|
| # | tying | matr. type | MFCC(20) | LDA(20) |
| 1 | global | diagonal | 2.62 | 2.11 |
| 8086 | no tying | diagonal | 2.52 | 1.98 |
| 1 | global | full | 2.46 | 1.99 |
| 8086 | | semi-tied | 2.26 | 1.90 |

4.2. Aurora2: noisy US-digit strings

The Aurora2 recognition systems (which are evolved from [5]) are trained either on the clean training set or on the multi-style noisy training set. Recognition results are given separately for the test sets A, B and C. The small recognition system (Tab. 3) consists of 234 mixtures with 0.9k Laplacians in total; the large system (Tab. 4) has 468 gender-dependent mixtures with 14k Laplacians. The Laplacian models either make use of a single globally tied diagonal deviation or of a single globally tied full deviation matrix. For best performance the LDA transformed features can be projected to 32 dimensional feature vectors; to be comparable to the original MFCC feature dimension results for LDA projections to 24 dimensions are also given.

In most cases the full deviation matrix with MFCC features performs better than LDA transformed features with diagonal deviations on comparable feature vector sizes. Interestingly, the effect of transforming the MFCC vectors using either a LDA matrix or a ML estimated full deviation matrix is somewhat contrary on test sets B and C: on set B LDA sometimes even degrades performance (with 24 dimensional vectors) but the full deviation matrix improves performance; on set C (channel mismatch) this situation is reverse; however, the combination of both matrices leads to a total performance improvement. The average WER reduction using the combination of both transformations with respect to the large diagonal deviation baseline systems is about 14% relative.

5. Conclusion

We have shown how the standard Laplacian model can be extended to deal with inter-feature dependencies explicitly by introducing full and semi-tied deviation matrices. A memory efficient procedure for training the new model can be constructed that makes use of a simple approximation to step functions. Significant performance improvements have been observed on two

Table 3: WER of small Aurora2 clean/multi systems: 0.9k dens.

| features | dev. | Test A | Test B | Test C | aver. |
|----------------------|-------|--------|--------|--------|-------|
| clean training | | | | | |
| MFCC(24) | diag. | 13.03 | 12.53 | 14.96 | 13.21 |
| | full | 11.91 | 11.53 | 14.36 | 12.24 |
| +LDA(24) | diag. | 12.37 | 12.47 | 14.81 | 12.88 |
| | full | 11.99 | 11.70 | 15.43 | 12.55 |
| +LDA(32) | diag. | 12.17 | 12.30 | 14.08 | 12.59 |
| | full | 11.77 | 11.52 | 14.53 | 12.20 |
| multi style training | | | | | |
| MFCC(24) | diag. | 11.00 | 11.34 | 13.46 | 11.62 |
| | full | 9.70 | 9.97 | 13.60 | 10.58 |
| +LDA(24) | diag. | 10.11 | 10.97 | 13.36 | 11.09 |
| | full | 10.01 | 10.34 | 13.53 | 10.83 |
| +LDA(32) | diag. | 9.63 | 10.44 | 12.57 | 10.53 |
| | full | 9.38 | 9.60 | 13.17 | 10.22 |

Table 4: WER of large Aurora2 clean/multi systems: 14k dens.

| feat. | dev. | Test A | Test B | Test C | aver. |
|----------------------|-------|--------|--------|--------|-------|
| clean training | | | | | |
| MFCC(24) | diag. | 9.96 | 10.17 | 10.75 | 10.19 |
| | full | 8.90 | 9.09 | 10.46 | 9.28 |
| +LDA(24) | diag. | 9.47 | 10.30 | 10.06 | 9.91 |
| | full | 8.97 | 9.28 | 10.11 | 9.31 |
| +LDA(32) | diag. | 8.77 | 9.05 | 9.95 | 9.11 |
| | full | 8.31 | 8.27 | 10.22 | 8.66 |
| multi style training | | | | | |
| MFCC(24) | diag. | 7.96 | 8.45 | 10.21 | 8.60 |
| | full | 7.41 | 7.66 | 10.56 | 8.13 |
| +LDA(24) | diag. | 7.13 | 8.53 | 9.14 | 8.08 |
| | full | 7.02 | 8.02 | 9.24 | 7.84 |
| +LDA(32) | diag. | 6.97 | 7.61 | 9.15 | 7.65 |
| | full | 6.78 | 7.16 | 9.59 | 7.48 |

different digit recognition tasks using the proposed extensions. Future systems might be further improved by integrating the transformation matrix estimation into the density splitting process of model construction as proposed in [2].

6. References

- [1] H. Ney, A. Noll, "Acoustic-Phonetic Modeling in the Spicos System", IEEE Trans. Speech and Audio Processing, 2(2):312–319, April 1994.
- [2] M. Gales, "Semi-Tied Covariance Matrices for Hidden Markov Models", IEEE Trans. Speech and Audio Processing, 7(3):272–281, May 1999.
- [3] R. Gopinath, "Constrained Maximum Likelihood Modeling with Gaussian Distributions", Proceedings Broadcast news transcription and understanding workshop, 110–115, Lansdowne, 1998.
- [4] R. Haeb-Umbach, H. Ney, "Linear Discriminant Analysis for Improved Large Vocabulary Continuous Speech Recognition", Proceedings ICASSP, I 13–16, San Francisco, 1992.
- [5] M. Lieb, A. Fischer, "Experiments with the Philips ASR system on the Aurora Noisy Digits Database", Proceedings Eurospeech, 625–628, Aalborg, Denmark, 2001.