ON THE USE OF MATRIX DERIVATIVES IN INTEGRATED DESIGN OF DYNAMIC FEATURE PARAMETERS FOR SPEECH RECOGNITION

Rathinavelu Chengalvarayan

Lucent Speech Solutions
Lucent Technologies Inc.
2000 Lucent Lane, Naperville
Illinois 60566, USA
Email: rathi@lucent.com

ABSTRACT

In this work, an integrated approach to vector dynamic feature extraction is described in the design of a hidden Markov model (VVD-IHMM) based speech recognizer. The new model contains state-dependent, vector-valued weighting functions responsible for transforming static speech features into the dynamic ones. In this paper, the minimum classification error (MCE) is extended from the earlier formulation of VVD-IHMM that applies to a novel maximum-likelihood based training algorithm. The experimental results on alphabet classification demonstrate the effectiveness of the MCE-trained new model relative to VVD-IHMM using dynamic features that have been subject to optimization during MLE-training.

1. INTRODUCTION

In the past few years, use of the coefficients that measure dynamic changes in the spectra has resulted in demonstrated success in enhancing the performance of both speech recognition and speech parameter generation systems [6, 7, 9]. The time-varying linear filter coefficients [2] have been shown to provide an optimal construction of dynamic parameters from existing static ones. The statistical model, called the *vector-valued dynamic integrated HMM* (VVD-IHMM), which incorporates generalized dynamic speech features described in this paper is an extension of the *scalar-valed dynamic integrated HMM* (SVD-IHMM) [2]. The state-dependent weights to tranform static speech features into dynamic ones as explained in [2] were considered as scalar valued. The more general case of matrix-valued weighting coefficients is described in this paper. We show that our approach based on this technique is appropriate to model the dynamics of cepstra since the state-dependent regression is done independently for each dimension of the transformed cepstral space.

2. DYNAMIC FEATURES

Let $\mathcal{X} = \{\mathcal{X}_1, \mathcal{X}_2, \cdots, \mathcal{X}_T\}$ denote the vector sequence of static feature parameters having the length of T frames. The dynamic feature vector $\mathcal{Y}_t$ at time frame t is defined as a simple combination of the static features stretching over the interval f frames forward and b frames backward according to [3]

$$\mathcal{Y}_t \;=\; \sum_{k=-b}^{f} \mathcal{W}_{k,i}\mathcal{X}_{t+k}, \qquad 1 \le t \le T,$$

where $\mathcal{W}_{k,i}$ are the matrix-valued weighting coefficients ($\mathcal{W}$'s can be treated as a diagonal matrix for mathematical simplicity) associated with the Markov state i. To simplify the discussion, we assume that the static and dynamic features are statistically independent [2]. A Gaussian density associated with each VVD-IHMM state i (a total of N states) assumes the form

$$b_i(\mathcal{O}_t) \;=\; b_i(\mathcal{X}_t, \mathcal{Y}_t) \;=\; b_i(\mathcal{X}_t)b_i(\mathcal{Y}_t),$$

where $\mathcal{O}_t$ is the augmented feature parameters at frame t consists of both static and the dynamic feature vectors. In the above equation $b_i(\mathcal{X}_t)$ and $b_i(\mathcal{Y}_t)$ are d-dimensional unimodal Gaussian densities for static and dynamic features respectively, as

$$b_i(\mathcal{X}_t) = \frac{1}{(2\pi)^{\frac{d}{2}}|\Sigma_{x,i}|^{\frac{1}{2}}} \exp\left(\frac{-1}{2}[\mathcal{X}_t - \mu_{x,i}]^{Tr}\Sigma_{x,i}^{-1}[\mathcal{X}_t - \mu_{x,i}]\right)$$

$$b_i(\mathcal{Y}_t) = \frac{1}{(2\pi)^{\frac{d}{2}}|\Sigma_{y,i}|^{\frac{1}{2}}} \exp\left(\frac{-1}{2}[\mathcal{Y}_t - \mu_{y,i}]^{Tr}\Sigma_{y,i}^{-1}[\mathcal{Y}_t - \mu_{y,i}]\right)$$

where variables $\mathcal{X}$ and $\mathcal{Y}$ indicate the static and the dynamic features, respectively. The parameters $\mu_{x,i}$, $\mu_{y,i}$ are the state-dependent Gaussian mean vectors and $\Sigma_{x,i}$, $\Sigma_{y,i}$ are the state-dependent diagonal covariance matrices. Superscripts -1 and Tr denote matrix inversion and vector transposition with d being the dimension of the static and dynamic feature vectors.

3. MATRIX DERIVATIVES

This section reviews the important theorems in vector and matrix calculus. These theorems are necessary, in many situations for example in the areas of multivariate analysis and the linear model, to obtain the partial derivatives of a function with respect to a vector or matrix as a variable. Some general theorems, which are useful in statistical applications and closely related to this work, are demonstrated in the following subsections. Note that the theorems are not for general matrices, but each theorem is for a matrix with a specific form or pattern. A further information about the matrix calculus can be obtained from the literature [1, 5, 11, 13].

3.1. Theorem

Let q be a quadratic form in the n independent real variables $x_1, x_2, \cdots, x_n$ defined by $q(x) = x^{Tr}Ax$, where $A = [a_{ij}]$ is a $n \times n$ symmetric matrix of constants. Then

$$\frac{\partial q}{\partial x} \;=\; 2Ax.$$

Proof:

$$q(x) = \sum_{j=1}^{n}\sum_{i=1}^{n} x_i x_j a_{ij}.$$

The t-th element of $\partial q(x)/\partial x$ is $\partial q/\partial x_t$, and clearly

$$\begin{aligned}\left[\frac{\partial q}{\partial x_t}\right] &= \left[\sum_{j=1}^{n} x_j a_{tj} + \sum_{i=1}^{n} x_i a_{it}\right] \\ &= 2\left[\sum_{j=1}^{n} x_j a_{tj}\right] \\ &= 2Ax \quad (since A is symmetric.)\end{aligned}$$

3.2. Result

Let Q be defined by

$$Q = \Lambda^{Tr}\Sigma\Lambda$$

where Λ is $d \times 1$ vector given by

$$\sum_{n=1}^{N} W(n)X(n) - \mu,$$

Σ is $d \times d$ symmetric matrix, μ and $X(n)$ are $d \times 1$ vectors, $W(n)$ are $d \times d$ matrices, for different values of n. Then

$$\frac{\partial Q}{\partial W(k)} = 2\Sigma\Lambda X^{Tr}(k),$$

for $k = 0, 1, \cdots, N$.

Proof: The scalar function Q can be written as

$$\begin{aligned}Q &= \sum_{p=1}^{d}\sum_{q=1}^{d}[\Lambda_p\Lambda_q\Sigma_{pq}] \\ &= \sum_{p=1}^{d}\sum_{q=1}^{d}\left[\left(\sum_{r=1}^{d}\sum_{n=1}^{N} W_{pr}(n)X_r(n) - \mu_p\right)\right. \\ &\quad \left.\left(\sum_{r=1}^{d}\sum_{n=1}^{N} W_{qr}(n)X_r(n) - \mu_q\right)\Sigma_{pq}\right]\end{aligned}$$

and the uv-th element of $\partial Q/\partial W(k)$ is $\partial Q/\partial W_{uv}(k)$ for $k = 1, 2, \cdots, N$ and $u, v = 1, 2, \cdots, d$. Clearly

$$\begin{aligned}\frac{\partial Q}{\partial W_{uv}(k)} &= \sum_{q=1}^{d}\left(\sum_{r=1}^{d}\sum_{n=1}^{N} W_{qr}(n)X_r(n) - \mu_q\right)\Sigma_{uq}X_v(k) \\ &+ \sum_{p=1}^{d}\left(\sum_{r=1}^{d}\sum_{n=1}^{N} W_{pr}(n)X_r(n) - \mu_p\right)\Sigma_{pu}X_v(k) \\ &= X_v(k)\left[\sum_{q=1}^{d}\Lambda_q\Sigma_{uq} + \sum_{p=1}^{d}\Lambda_p\Sigma_{pu}\right] \\ &= 2X_v(k)\underbrace{\sum_{z=1}^{d}\Lambda_z\Sigma_{uz}}_{\xi_u}\end{aligned}$$

Now, the partial derivative of $W(k)$ matrix with respect to the scalar Q can be expressed, element by element, as

$$\begin{aligned}\frac{\partial Q}{\partial W(k)} &= 2\begin{bmatrix} \xi_1X_1(k) & \xi_1X_2(k) & \cdots & \xi_1X_d(k) \\ \xi_2X_1(k) & \xi_2X_2(k) & \cdots & \xi_2X_d(k) \\ \vdots & \vdots & \ddots & \vdots \\ \xi_dX_1(k) & \xi_dX_2(k) & \cdots & \xi_dX_d(k) \end{bmatrix} \\ &= 2\begin{bmatrix} \xi_1 \\ \xi_2 \\ \vdots \\ \xi_d \end{bmatrix}\begin{bmatrix} X_1(k) & X_2(k) & \cdots & X_d(k) \end{bmatrix} \\ &= 2\begin{bmatrix} \sum_{z=1}^{d}\Lambda\Sigma \\ \sum_{z=1}^{d}\Lambda\Sigma \\ \vdots \\ \sum_{z=1}^{d}\Lambda\Sigma \end{bmatrix}X^{Tr}(k) \\ &= 2\Sigma\Lambda X^{Tr}(k)\end{aligned}$$

for $k = 1, 2, \cdots, N$.

4. DISCRIMINATIVE TRAINING

The approach we employed in previous work to the parameter estimation problem associated with the integrated HMM has been built on the basis of the maximum likelihood (MLE) principle [3, 4]. Use of an entirely new theoretical principle, the minimum classification error (MCE) principle, to design the integrated HMM is the subject of the current section [8, 10]. In the supervised training mode which we assume, each training token (an augmented l-th data sequence having the length of T^l frames $\mathcal{X}^l_{T^l}, \mathcal{Y}^l_{T^l})$) is known to belong to one of $\mathcal{K}$ classes $\{C^j\}_{j=1}^{\mathcal{K}}$. The goal of the MCE-training is to find the classifier parameter set, denoted by $\Phi = \{\Phi^j\}_{j=1}^{\mathcal{K}}$, such that the probability of misclassifying any $\mathcal{O}^l$ is minimized and the resulting Φ gives the optimal solution of the classifier.

The first step in the formulation of the objective function is to choose an appropriate discriminant function $g_\kappa(\mathcal{O}^l, \Phi)$ according to the following decision rule for classification:

$$C(\mathcal{O}^l) = C^\kappa, \ if \ g_\kappa(\mathcal{O}^l, \Phi) = \max_j \ g_j(\mathcal{O}^l, \Phi)$$

where $C(.)$ is the class associated with the test data $\mathcal{O}^l$ as determined by the classifier. In our implementation of the integrated HMM, we choose the most likely (optimal) state path traversing the Markov model as the basis for defining the discriminant function. The log-likelihood score of the input utterance $\mathcal{O}^l$ along the optimal state sequence $\Theta^\kappa = \{\theta^\kappa_1, \theta^\kappa_2 \cdots, \theta^\kappa_{T^l}\}$ for the model associated with the κth class Φ^κ can be written as

$$g_\kappa(\mathcal{O}^l, \Phi) = \sum_{t=1}^{T^l} \log b_{\theta^\kappa_t}(\mathcal{O}^l_t)$$

where $b_{\theta^\kappa_t}(\mathcal{O}^l_t)$ is the probability of generating the feature vector $\mathcal{O}^l_t$ at time t in state θ^κ_t by the model for class κth, and T^l is the number of frames of the lth observation sequence.

In our implementation of the integrated HMM, we assume uncorrelatedness between the static features and the generalized dynamic features. Hence, the output probability takes the form

$$b_{\theta^\kappa_t}(\mathcal{O}^l_t) = b_{\theta^\kappa_t}(\mathcal{X}^l_t, \mathcal{Y}^l_t) = b_{\theta^\kappa_t}(\mathcal{X}^l_t)b_{\theta^\kappa_t}(\mathcal{Y}^l_t),$$

Given a discriminant function, a misclassification measure for an input training utterance $\mathcal{O}^l$ from class κ can be de-

fined as follows to quantify the classification behavior:

$$d_\kappa(\mathcal{O}^l,\Phi) = -g_\kappa(\mathcal{O}^l,\Phi)+\log\left[\frac{1}{\mathcal{K}-1}\sum_{j\neq\kappa}e^{g_j(\mathcal{O}^l,\Phi)\eta}\right]^{\frac{1}{\eta}}$$

where η is a positive number and $\mathcal{K}$ is the total number of classes. $d_\kappa(\mathcal{O}^l,\Phi)$ above is a quantity that indicates the degree of confusion between the correct class and the other competing classes for a given input utterance $\mathcal{O}^l$. When η approaches ∞, the misclassification measure becomes

$$\begin{aligned} d_\kappa(\mathcal{O}^l,\Phi) &\Rightarrow -g_\kappa(\mathcal{O}^l,\Phi)+\max_{j\neq\kappa}g_j(\mathcal{O}^l,\Phi) \\ &= -g_\kappa(\mathcal{O}^l,\Phi)+g_i(\mathcal{O}^l,\Phi), \end{aligned}$$

where C^i is the most confusable class. Clearly, a positive value of $d_\kappa(\mathcal{O}^l,\Phi)$ indicates a misclassification and a negative value of $d_\kappa(\mathcal{O}^l,\Phi)$ implies a correct decision. Given a misclassification measure, we further define a smoothed loss function for each class κ:

$$\Upsilon_\kappa(\mathcal{O}^l,\Phi) = \frac{1}{1+e^{-\rho d_\kappa(\mathcal{O}^l,\Phi)}},\quad \rho>0$$

which approximates the classification error count. Finally, given a loss function defined for each class, we define the overall loss function for the entire classifier as

$$\Upsilon(\mathcal{O}^l,\Phi) = \sum_{\kappa=1}^{\mathcal{K}}\Upsilon_\kappa(\mathcal{O}^l,\Phi)\delta[\mathcal{O}^l\in C^\kappa]$$

where $\delta[\xi]$ is the Kronecker indicator function of a logic expression ξ that gives value 1 if the value of ξ is true and value 0 otherwise. The loss function $\Upsilon(\mathcal{O}^l,\Phi)$ is minimized, each time a training token $\mathcal{O}^l$ is presented, by adaptively adjusting the parameter set Φ according to

$$\Phi_{l+1} = \Phi_l-\epsilon\nabla\Upsilon(\mathcal{O}^l,\Phi_l),$$

where Φ_l is the parameter set at the lth iteration, $\nabla\Upsilon(\mathcal{O}^l,\Phi_l)$ is the gradient of the loss function for training sample $\mathcal{O}^l$. ϵ is a small positive learning constant. Let ϕ^j denote a parameter associated with model j, then in the case of token-by-token training, we can write the gradient as

$$\begin{aligned} \frac{\partial\Upsilon(\mathcal{O}^l,\Phi)}{\partial\phi^j} &= \frac{\partial}{\partial\phi^j}\left(\sum_{\kappa=1}^{\mathcal{K}}\Upsilon_\kappa(\mathcal{O}^l,\Phi)\delta[\mathcal{O}^l\in C^\kappa]\right) \\ &= \frac{\partial}{\partial\phi^j}\Upsilon_\kappa(\mathcal{O}^l,\Phi) \\ &= \frac{\partial\Upsilon_\kappa(\mathcal{O}^l,\Phi)}{\partial d_\kappa(\mathcal{O}^l,\Phi)}\frac{\partial d_\kappa(\mathcal{O}^l,\Phi)}{\partial g_j(\mathcal{O}^l,\Phi)}\frac{\partial g_j(\mathcal{O}^l,\Phi)}{\partial\phi^j} \\ &= \psi_j\frac{\partial g_j(\mathcal{O}^l,\Phi)}{\partial\phi^j} \end{aligned}$$

where the quantity ψ_j is defined as

$$\psi_j=\begin{cases} -\rho\Upsilon_\kappa(\mathcal{O}^l,\Phi)[1-\Upsilon_\kappa(\mathcal{O}^l,\Phi)] & \text{if } j=\kappa \\ \rho\Upsilon_\kappa(\mathcal{O}^l,\Phi)[1-\Upsilon_\kappa(\mathcal{O}^l,\Phi)]\frac{e^{\eta g_j(\mathcal{O}^l,\Phi)}}{\sum_{\ell\neq\kappa}e^{\eta g_\ell(\mathcal{O}^l,\Phi)}} & \text{if } j\neq\kappa \end{cases}$$

In the remaining of this section, class index j will be omitted for clarity of presentation. By using the matrix derivative result (3.2), the partial derivative of $\Upsilon(\mathcal{O}^l,\Phi)$ with respect to each $W_i(k)$ is given by

$$\begin{aligned} \frac{\partial\Upsilon(\mathcal{O}^l,\Phi)}{\partial W_i(k)} &= \psi\frac{\partial g(\mathcal{O}^l,\Phi)}{\partial W_i(k)} \\ &= \psi\frac{\partial}{\partial W_i(k)}\left(\sum_{t=1}^{T^l}\log b_{\theta_t}(\mathcal{O}_t^l)+\sum_{t=1}^{T^l-1}\log a_{\theta_t\theta_{t+1}}\right) \\ &= \psi\sum_{t\in T_i^l}\frac{1}{b_i(\mathcal{O}_t^l)}\frac{\partial}{\partial W_i(k)}b_i(\mathcal{O}_t^l) \\ &= \psi\sum_{t\in T_i^l}\frac{1}{b_i(\mathcal{O}_t^l)}\frac{\partial}{\partial W_i(k)}\left(b_i(\mathcal{X}_t^l)b_i(\mathcal{Y}_t^l)\right) \\ &= \psi\sum_{t\in T_i^l}\frac{b_i(\mathcal{X}_t^l)}{b_i(\mathcal{O}_t^l)}\frac{\partial b_i(\mathcal{Y}_t^l)}{\partial W_i(k)} \\ &= \psi\sum_{t\in T_i^l}\frac{b_i(\mathcal{X}_t^l)b_i(\mathcal{Y}_t^l)}{b_i(\mathcal{O}_t^l)}\frac{\partial}{\partial W_i(k)} \\ &\quad\left(-\frac{1}{2}[\mathcal{Y}_t^l-\mu_{y,i}]^{Tr}\Sigma_{y,i}^{-1}[\mathcal{Y}_t^l-\mu_{y,i}]\right) \\ &= -\psi\sum_{t\in T_i^l}\Sigma_{y,i}^{-1}[\mathcal{Y}_t^l-\mu_{y,i}][\mathcal{X}_{t+k}^l]^{Tr}, \end{aligned}$$

for $i=1,2,\cdots,N$, $k=-b,-b+1,\cdots,f$ and the set T_i^l includes all the time indices such that the state index of the state sequence at time t belongs to state ith in the Markov chain, i.e. $T_i^l=\{t|\theta_t=i\}$, $1\le i\le N$, $1\le t\le T^l$.

The following gradient equations are obtained by using the matrix calculus techniques [1] and by computing the partial derivatives of $\Upsilon(\mathcal{O}^l,\Phi)$ with respect to each integrated HMM parameter for a given training token $\mathcal{O}^l$ belonging to class κth:

$$\begin{aligned} \frac{\partial\Upsilon(\mathcal{O}^l,\Phi)}{\partial\mu_{x,i}} &= \psi\sum_{t\in T_i^l}\Sigma_{x,i}^{-1}\Lambda_{x,i,t} \\ \frac{\partial\Upsilon(\mathcal{O}^l,\Phi)}{\partial\mu_{y,i}} &= \psi\sum_{t\in T_i^l}\Sigma_{y,i}^{-1}\Lambda_{y,i,t} \\ \frac{\partial\Upsilon(\mathcal{O}^l,\Phi)}{\partial\tilde{\Sigma}_{x,i}} &= 0.5\psi\sum_{t\in T_i^l}[\Sigma_{x,i}^{-1}\Lambda_{x,i,t}\Lambda_{x,i,t}^{Tr}-I] \\ \frac{\partial\Upsilon(\mathcal{O}^l,\Phi)}{\partial\tilde{\Sigma}_{y,i}} &= 0.5\psi\sum_{t\in T_i^l}[\Sigma_{y,i}^{-1}\Lambda_{y,i,t}\Lambda_{y,i,t}^{Tr}-I] \end{aligned}$$

where I denotes $d\times d$ identity matrix, $\Lambda_{x,i,t}=\mathcal{X}_t^l-\mu_{x,i}$, $\Lambda_{y,i,t}=\mathcal{Y}_t^l-\mu_{y,i}$, For easier implementation, the constrained parameters are transformed to an unconstrained domain and the gradient is computed with respect to the transformed parameters $\tilde{\Sigma}_{x,i}=\log\Sigma_{x,i}$ and $\tilde{\Sigma}_{y,i}=\log\Sigma_{y,i}$.

5. CLASSIFICATION EXPERIMENTS

The experiments conducted to evaluate the various integrated HMMs are aimed at recognizing the 26 letters in the

Type of model	MLE-model	MCE-model
Conventional HMM	80.11%	84.68%
SVD-IHMM	81.55%	86.16%
VVD-IHMM	82.57%	87.45%
VVD-IHMM*	82.57%	88.39%

Table 1. TI 26-alphabet classification rate as a function of the model type.

English alphabet, contained in the TI46 isolated word corpus. The speaker-independent training set consists of 10 tokens per word from two male and two female speakers (m1, m2, f1 and f2). The remaining 16 tokens per word for each of the above four speakers is used as test data. The preprocessor produces a vector of 13 Mel-frequency cepstral coefficients (MFCCs) for every 10 msec throughout the signal. The augmented feature vectors used for the benchmark HMM consist of 26-elements, with 13 cepstrum coefficients and 13 delta cepstra. For the integrated HMMs, only the static feature vectors are used as the raw data to the recognizer, which constructs the dynamic feature parameters internally within the recognizer. To be consistent with the conventional delta parameter techniques, the window variables f, b are set to 2 and the matrix $W_{k,i}$ is constrained to be diagonal.

Each word is represented by a single left-to-right, three-state HMM (no skips), with single Gaussian state observation densities. The covariance matrices in all the states of all the models are diagonal and are not tied. All transition probabilities are uniformly set to 0.5 and are not learned during the training process. The conventional HMM models are trained from training data using five-iterations of the MLE-training with single mixture for each state in the HMMs [12]. The scalar-valued dynamic integrated HMM (SVD-IHMM) are trained using five-iterations of the MLE-algorithm with non-linear type constraint [2]. The vector-valued dynamic integrated HMM (VVD-IHMM) are trained according to the training procedure outlined in [3]. The MLE-trained models are used as the initial model for the ensuing MCE-training step. The conventional HMM models and SVD-IHMM are discriminatively trained using five iterations of gradient probabilistic descent based MCE-algorithm [2]. The VVD-IHMM* is trained using five iterations of MCE-procedure outlined in the previous section. During MCE-training for VVD-IHMM, we updated only the state-dependent mean and variance parameters and the vector-valued dynamic feature parameters are kept constant during MCE-training process, but they are trained using non-linear type contraint based MLE-training [4].

The experimental results are summarized in Table 1. We observe from Table 1 that all the IHMMs are superior to the conventional HMM. The SVD-IHMM based classifier produces 86.16% accuracy with an error rate reduction of 10% compared with the convention HMM classifier (84.68%). From the final classifier based on VVD-IHMM*, which incorporated vector-valued dynamic weighting functions, the best classification results have been obtained (88.39%). The recognition rate using the VVD-IHMM* improved from 84.68% (conventional MCE-trained HMM) to 88.39% which translates to 24% error rate reduction. It also represents a 16% error rate reduction compared with SVD-IHMM. Among all four types of the model evaluated, the MCE-trained VVD-IHMM* performs better than the other models.

6. CONCLUSIONS

The state-dependent weights to tranform static speech features into dynamic ones as explained in [2] were considered as scalar valued. The more general case of matrix-valued weighting coefficients is developed and evaluated using MCE-training algorithm. The best error rate reduction of 24% is obtained using the MCE-trained VVD-IHMM*, tested on a TI alphabet classification task, relative to conventional HMM. Compared across all four MCE-trained classifiers, VVD-IHMM* produced the lowest error rate and is the new efficient way of describing the dynamic characteristics of speech cepstra.

REFERENCES

[1] R. Chengalvarayan, "Matrix-calculus techniques for numerical solution of a hierarchical nonstationary model with applications to speech modeling and recognition", *M.A.Sc. Thesis*, University of Waterloo, Ontario, Canada, 1992, pp. 70-80.

[2] R. Chengalvarayan and L. Deng, "Use of generalized dynamic feature parameters for speech recognition: maximum likelihood and minimum classification error approaches", *IEEE Proc. ICASSP*, pp. 373-376, 1995.

[3] R. Chengalvarayan, "Use of vector-valued dynamic weighting coefficients for speech recognition: Maximum likelihood approach", *Proc. EUROSPEECH*, pp. 501-504, 1997.

[4] R.Chengalvarayan, "Discriminative hidden Markov models using vector-valued dynamic weighting parameters for alphabet recognition", *Proc. on Systemics, Cybernetics and Informatics*, Vol. 5, pp. 37-40, 2000.

[5] P. S. Dwyer, "Some applications of matrix derivatives in multivariate analysis", *Journal of American Statistical Association*, Vol. 62, pp. 607-625, 1967.

[6] V. N. Gupta, M. Lennig and P. Mermelstein, "Integration of acoustic information in a large vocabulary word recognizer", *IEEE Proc. ICASSP*, Vol. 2, pp. 697-700, 1987.

[7] B. Hanson and T. Applebaum, "Robust speaker-independent word recognition using static, dynamic and acceleration features: experiments with lombard and noisy speech", *IEEE Proc. ICASSP*, pp. 857-860, 1991.

[8] B. H. Juang, W. Chou and C. H. Lee, "Minimum classification error rate methods for speech recognition", *IEEE Transactions on Speech and Audio Processing*, Vol. 5, No.3, pp. 257-265, 1997.

[9] C. H. Lee, L. Rabiner, R. Pieraccini, and J. Wilpon, "Acoustic modeling for large vocabulary speech recognition", *Computer Speech and Language*, Vol. 4, pp. 127-165, 1990.

[10] E. McDermott and S. Katagiri, "String-Level MCE for Continuous Phoneme Recognition", *Proc. EUROSPEECH*, pp. 123-126, 1997.

[11] R. P. McDonald and H.Swaminathan, "A simple matrix calculus with applications to multivariate analysis", *General Systems*, Vol. 18, pp. 37-54, 1973.

[12] L. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition", *The IEEE Proceedings*, Vol. 77, pp. 257-285, 1989.

[13] G. S. Rogers, "Matrix Derivatives", *Lecture notes in statistics*, Marcel Dekker, New York, Vol. 2, pp. 41-50, 1980.