Strong-Sense Class-Dependent Features For Statistical Recognition

Mohamed Kamal Omar, Mark Hasegawa-Johnson

Department of Electrical And Computer Engineering,
University of Illinois at Urbana-Champaign,
Urbana, IL 61801

omar,jhasegaw@uiuc.edu

Abstract

In statistical classification and recognition problems with many classes, it is commonly the case that different classes exhibit wildly different properties. In this case it is unreasonable to expect to be able to summarize these properties by using features designed to represent all the classes. In contrast, features should be designed to represent subsets that exhibit common properties without regard to any class outside this subset. The value of these features for classes outside the subset may be meaningless, or simply undefined. The main problem, due to the statistical nature of the recognizer, is how to compare likelihoods conditioned on different sets of features to decode an input pattern.

This paper introduces a class-dependent feature design approach that can be integrated with any probabilistic model. This approach avoids the need of having a conditional probabilistic model for each class and feature type pair, and therefore decreases the computational and storage requirements of using heterogeneous features. We present in this paper an algorithm to calculate the class-dependent features that minimize an estimate of the relative entropy between the conditional probabilistic model and the actual conditional probability density function (PDF) of the features of each class.

We apply our approach to a hidden Markov model (HMM) automatic speech recognition (ASR) system. We use a non-linear class-dependent volume-preserving transformation of the features to minimize the objective function. Using this approach, we achieved 2% improvement in phoneme recognition accuracy compared to the baseline system. Our approach shows also improvement in recognition accuracy compared to previous class-dependent linear features transformation.

1. Introduction

The class-dependent features can be looked at as a method of dimensionality reduction in classification [1]. Unlike other methods of dimensional reduction, it is based on sufficient statistics and results in no theoretical loss of performance. Statistical classifiers lose information necessary for classification and recognition in two ways. The first is due to reducing the given data to a set of features, and the second is due to approximating the true joint PDFs of the features. The former loss reduces as the dimensionality of the features is increased, while the latter increases as the dimensionality of the features is increased. Class-dependent features avoid this compromise by allowing more information to be kept for a given maximum feature dimension. This is clearly at the expense of increasing the computational requirements of the system. Class-dependent features are motivated by the fact that different classes have different salient characteristics that may require different features.

Many recent speech recognition systems use class-dependent feature streams to achieve more robustness and better performance [2]. Using different feature streams within each recognizer allows the overall system to benefit from the ability of these streams to reveal complementary information about the original speech signal. The main problem, due to the statistical nature of the recognizer, is how to compare likelihoods conditioned on different sets of features to decode a given utterance [3]. The previous approaches to this problem include model-based approaches, and feature-based approaches. In model based approaches, the problem is solved by either completely abandoning the statistical structure of the recognizer, or by adding extra reference models that have no physical meaning but are used to normalize the likelihoods to be comparable statistically [3]. The main problem with the latter approach is how to train these reference models. They are synthetic entities that have no physical meaning at all, so there have been a variety of suggestions to train these models. They range from taking all other phones in the phone set to train the reference model to taking a very small set of similar phones in the phone set. The feature-based approach restricted the class-dependent features to features generated by class-dependent linear transforms of an original set of features [4].

In the weak sense class-dependency, features have observable values for all classes, but the features and some class variables are conditionally independent given a set of classes [5]. This increases the computational and the storage requirements of the system, and results in the introduction of meaningless models that degrade the performance of the recognizer. Features are said to be class-dependent in the strong sense if they are assumed to be observable only for one class or cluster of classes but undefined for the rest of the classes. In this paper, a non-linear strong-sense class-dependent feature transformation for pattern recognition is described. It is applied to an HMM speech recognizer. The feature streams are optimized to minimize an estimate of the relative entropy between the actual conditional likelihood and its estimation based on the model. We will use here the notion of class-dependent features for ASR to represent using different features for different phonemes or different clusters of phonemes that are constructed using some criterion. In the next section, the problem is formulated and a solution based on volume-preserving maps is introduced. The criterion for optimizing the class-dependent features is discussed in section 3. An iterative algorithm is described in section 4 to jointly estimate the parameters of the feature transform and the parameters of the model. Finally, recognition experiments are described in section 5.

2. Problem Formulation

The Bayes classification rule minimizes the probability of error if the underlying distribution of the data is known. In its original format, it assumes that the same features are used for all classes; The Bayes classification rule for a set of classes c_i for $i = 1, 2, \cdots, J$ is

$$\hat{c} = \arg \max_{c \in \{1,\cdots,J\}} P_{C|X}(c|x), \quad (1)$$

where J is the number of classes, x is the observation vector, and $P_{C|X}(c|x)$ is the *a posteriori* probability of the classes. This maximization can be reduced to

$$\hat{c} = \arg \max_{c \in \{1,\cdots,J\}} P_{X|C}(x|c)P(c). \quad (2)$$

Let us now define a set of functions $\{f_i(.)\}_{i=1}^{J}$ such that $y_i = f_i(x)$ is an arbitrary one-to-one map of the random vector X in $\Re^n$ to Y_i in $\Re^n$.

The relation between the joint class-conditional probability of X and Y_i is

$$P_{Y_i|C}(f_i(x)|c_i) = \frac{P_{X|C}(x|c_i)}{|det(\frac{\partial f_i}{\partial x})|}, \quad (3)$$

where $det(\frac{\partial f_i}{\partial x})$ is the determinant of the Jacobian matrix of the map $f_i(.)$ [10].

Therefore, the Bayesian classification rule for the classifiers that use a set of class-dependent features, $\{y_i\}_{i=1}^{J}$ becomes

$$\hat{c} = \arg \max_{c \in \{1,\cdots,J\}} P_{Y_i|C}(f_i(x)|c)P(c)|det(\frac{\partial f_i}{\partial x})|. \quad (4)$$

Equation 4 shows that we can design strong-sense class-dependent features for any statistical recognition or classification system by taking the determinant of the Jacobian matrix into consideration in the decision rule.

An important special case that simplifies the decision rule is volume-preserving maps.

Definition: A C^∞ map $f : S_x \to S_y$ where $S_x \subset \Re^n$ and $S_y \subset \Re^n$ is said to be volume-preserving if and only if $\left|det\left(\frac{\partial f(x)}{\partial x}\right)\right| = 1 \; \forall x \in S_x$.

Therefore, the Bayesian classification rule for the classifiers that use a set of class-dependent features, $\{y_i\}_{i=1}^{J}$ generated using a set of volume-preserving maps becomes

$$\hat{c} = \arg \max_{c \in \{1,\cdots,J\}} P_{Y_i|C_i}(f_i(x)|c_i)P(c_i). \quad (5)$$

3. Optimization of The Class-Dependent Features

Since we do not have the actual joint class-conditional PDFs, a parametric model of these PDFs is usually used in the statistical recognition system. We choose the class-dependent features transform and the model parameters such that they minimize the relative entropy between the actual joint class-conditional PDFs and their parametric model for each class.

Proposition: Let $y_i = f_i(x)$ for $i = 1, \cdots, J$ be arbitrary one-to-one C^∞ volume-preserving maps of the random vector X in $\Re^n$ to Y_i in $\Re^n$, and let $\hat{P}_{\Lambda_i}(y_i|c_i)$ be the parametric model of the conditional PDF. The map $f_i^*(.)$ and the set of parameters Λ_i^* jointly minimize the relative entropy between the hypothesized and the true conditional likelihoods of Y_i if and only if they also maximize the expected log likelihood based on the model, $E_{P(Y_i|C_i)}[\log \hat{P}_{\Lambda_i}(y_i|c_i)]$.

Proof: The expression for the relative entropy after an arbitrary transformation, $y_i = f_i(x)$, of the input random vector X in $\Re^n$, is

$$\begin{aligned} R(P(Y_i|c_i), \hat{P}_{\Lambda_i}(Y_i|c_i)) &= -H(P(Y_i|c_i)) \\ &\quad -E_{P(Y_i|c_i)}\left[\log\left(\hat{P}(Y_i|c_i)\right)\right], \end{aligned} \quad (6)$$

where $H(P(Y_i|c_i))$ is the conditional differential entropy of the random vector Y_i [9].

The relation between the output conditional differential entropy and the input conditional differential entropy is in general,

$$\begin{aligned} H(P(Y_i|c_i)) &\leq H(P(X|c_i)) \\ &\quad + \int_{\Re^n} P(x|c_i) \log\left(\left|det\left(\frac{\partial f_i(x)}{\partial x}\right)\right|\right) dx, \end{aligned} \quad (7)$$

where $P(x|c_i)$ is the conditional probability density function of the random vector X, for an arbitrary transformation, $y_i = f_i(x)$, of the random vector X in $\Re^n$, with equality if $f_i(x)$ is invertible [10].

Therefore, for a volume-preserving map $y_i = f_i(x)$, the relative entropy can be written as

$$\begin{aligned} R(P(Y_i|c_i), \hat{P}_{\Lambda_i}(Y_i|c_i)) &= -H(P(X|c_i)) \\ &\quad -E_{P(Y_i|c_i)}\left[\log \hat{P}_{\Lambda_i}(Y_i|c_i)\right]. \end{aligned} \quad (8)$$

Equation 8 proves the proposition.

The problem of maximizing $E_{P(Y_i|C_i)}[\log \hat{P}_{\Lambda_i}(y_i|c_i)]$ can be solved using efficient algorithms based on the incremental expectation maximization (EM) algorithm. Our approach generalizes previous approaches to class-dependent feature transform for speech recognition in two ways. First, the feature transform is not necessarily linear. Second, it is class-dependent in the strong sense: the conditional PDF of each features stream is calculated for the corresponding class only.

It should be noted that all linear maps designed to improve the satisfaction of the features of a given model are special cases of the proposition, as any linear map is equivalent to a linear volume-preserving map multiplied by a scalar.

4. Implementation of The Maximum Likelihood Approach

In the previous section, we showed that by using a volume-preserving map, the problem of minimizing the relative entropy between the actual class-conditional PDFs and their parametric models is reduced to maximizing the likelihood of the training data in the new feature space. In this section, we use a symplectic map to generate the new set of features.

4.1. Symplectic Maps

Symplectic maps are volume-preserving maps that can be represented by scalar functions [11]. This interesting result allows us to jointly optimize the parameters of the symplectic map and

the model parameters using the EM algorithm or one of its incremental forms [6].

Let $x = (x_1, x_2)$, and $y = (y_1, y_2)$, with $x_1, x_2, y_1, y_2 \in \Re^{\frac{n}{2}}$, then any reflecting symplectic map can be represented by [11]

$$y_1 = x_1 - \frac{\partial V(x_2)}{\partial x_2}, \quad (9)$$

$$y_2 = x_2 - \frac{\partial T(y_1)}{\partial y_1}, \quad (10)$$

where $V(\cdot)$ and $T(\cdot)$ are two arbitrary scalar functions. We use two three-layer feed-forward neural networks to get a good approximation of these scalar functions.

$$V(u, A, C) = \sum_{j=1}^{M} c_j S(a_j u), \quad (11)$$

$$T(u, B, D) = \sum_{j=1}^{M} d_j S(b_j u), \quad (12)$$

where $S(.)$ is a nonlinear function like sigmoid or hyperbolic tangent, a_j is the jth row of the $M \times n$ matrix A, and c_j is the jth element of the $M \times 1$ vector C, b_j is the jth row of the $M \times n$ matrix B, and d_j is the jth element of the $M \times 1$ vector D. The parameters of these two neural networks and the parameters of the model are jointly optimized to maximize the likelihood of the training data.

4.2. Joint Optimization of The Map and Model Parameters

We will describe here an iterative algorithm to estimate the parameters of the symplectic map and a hidden Markov model (HMM) for speech recognition. Using the EM algorithm, the auxiliary function to be maximized is

$$Q(\Phi^k, \Phi^{k+1}) = E_\xi[\log P(y, \zeta|\Phi^{k+1})|y, \Phi^k], \quad (13)$$

where $\zeta \in \xi$ is the state sequence corresponding to the sequence of observations $x \in \Re^{n \times T}$ that are transformed to the sequence $y \in \Re^{n \times T}$, T is the sequence length in frames, $\Phi^k = (\Lambda^k, W^k)$ is the set of the recognizer parameters and the symplectic parameters at iteration k of the algorithm.

The updating equations for the HMM parameters are not affected by the introduction of the feature transform, and therefore will not be given here.

We will assume that the recognizer models the conditional PDF of the observation as a mixture of diagonal-covariance Gaussians and therefore

$$\frac{\partial Q(\Phi^k, \Phi^{k+1})}{\partial y_j} = \sum_{i=1}^{N} \sum_{m=1}^{K} \frac{P(y^i, m|\Phi^k)}{P(y^i|\Phi^k)} \frac{(\mu_{mj} - y_j^i)}{\sigma_{mj}^2}, \quad (14)$$

where μ_{mj}, and σ_{mj}^2 are the mean and the variance of the jth element of the mth PDF respectively, N is the number of frames in the training data, and K is the total number of Gaussian models.

Let the nonlinearity, $S(.)$, in the neural networks be hyperbolic tangent functions. Starting with A and B, to update the values of the symplectic parameters a_{qr} and b_{qr} for $q = 1, 2, \cdots, M$, and for $r = 1, 2, \cdots, \frac{n}{2}$, we have to calculate the partial derivative of the auxiliary function with respect to these parameters. These partial derivatives are related to the partial derivatives of the auxiliary function with respect to the features by the following relation

$$\frac{\partial Q(\Phi^k, \Phi^{k+1})}{\partial a_{qr}} = \sum_{j=1}^{\frac{n}{2}} \frac{\partial Q(\Phi^k, \Phi^{k+1})}{\partial y_{1j}} \frac{\partial y_{1j}}{\partial a_{qr}} + \sum_{j=1}^{\frac{n}{2}} \frac{\partial Q(\Phi^k, \Phi^{k+1})}{\partial y_{2j}} \frac{\partial y_{2j}}{\partial a_{qr}}, \quad (15)$$

and

$$\frac{\partial Q(\Phi^k, \Phi^{k+1})}{\partial b_{qr}} = \sum_{j=1}^{\frac{n}{2}} \frac{\partial Q(\Phi^k, \Phi^{k+1})}{\partial y_{2j}} \frac{\partial y_{2j}}{\partial b_{qr}}, \quad (16)$$

where

$$\frac{\partial y_{1j}}{\partial a_{qr}} = \begin{cases} 2x_{2r} \sum_{h=1}^{M} \left(c_h a_{hj} S(a_h x_2)[1 - S^2(a_h x_2)]\right) & \text{for } r \neq j \\ 2x_{2r} \sum_{h=1}^{M} \left(c_h a_{hj} S(a_h x_2)[1 - S^2(a_h x_2)]\right) & \\ -c_q[1 - S^2(a_q x_2)] \quad \text{for } r = j \end{cases} \quad (17)$$

$$\frac{\partial y_{2j}}{\partial a_{qr}} = \sum_{k=1}^{\frac{n}{2}} \frac{\partial y_{1k}}{\partial a_{qr}} \frac{\partial y_{2j}}{\partial y_{1k}}, \quad (18)$$

$$\frac{\partial y_{2j}}{\partial y_{1k}} = -\sum_{h=1}^{M} \left(d_h b_{hj} b_{hk} S(b_h y_1)[1 - S^2(b_h y_1)]\right), \quad (19)$$

and

$$\frac{\partial y_{2j}}{\partial b_{qr}} = \begin{cases} 2y_{1r} \sum_{h=1}^{M} \left(c_h b_{hj} S(b_h y_1)[1 - S^2(b_h x_2)]\right) & \text{for } r \neq j \\ 2y_{1r} \sum_{h=1}^{M} \left(c_h b_{hj} S(b_h y_1)[1 - S^2(b_h x_2)]\right) & \\ -d_q[1 - S^2(b_q y_1)] \quad \text{for } r = j \end{cases} \quad (20)$$

For C and D, to update values of the symplectic parameters c_q and d_q for $q = 1, 2, \cdots, M$, we have to calculate the partial derivative of the auxiliary function with respect to these parameters. These partial derivatives are related to the partial derivatives of the auxiliary function with respect to the features by the following relation

$$\frac{\partial Q(\Phi^k, \Phi^{k+1})}{\partial c_q} = \sum_{j=1}^{\frac{n}{2}} \frac{\partial Q(\Phi^k, \Phi^{k+1})}{\partial y_{1j}} \frac{\partial y_{1j}}{\partial c_q} + \sum_{j=1}^{\frac{n}{2}} \frac{\partial Q(\Phi^k, \Phi^{k+1})}{\partial y_{2j}} \frac{\partial y_{2j}}{\partial c_q}, \quad (21)$$

and

$$\frac{\partial Q(\Phi^k, \Phi^{k+1})}{\partial d_q} = \sum_{j=1}^{\frac{n}{2}} \frac{\partial Q(\Phi^k, \Phi^{k+1})}{\partial y_{2j}} \frac{\partial y_{2j}}{\partial d_q}, \quad (22)$$

where

$$\frac{\partial y_{1j}}{\partial c_q} = a_{qj}[1 - S^2(a_q x_2)], \quad (23)$$

$$\frac{\partial y_{2j}}{\partial c_q} = \sum_{k=1}^{\frac{n}{2}} \frac{\partial y_{1k}}{\partial c_q} \frac{\partial y_{2j}}{\partial y_{1k}}, \quad (24)$$

and

$$\frac{\partial y_{2j}}{\partial d_q} = b_{qj}[1 - S^2(b_q y_1)]. \quad (25)$$

Using Equations 14 to 25, the values of the symplectic map parameters can be estimated in each iteration using a gradient-based optimization algorithm.

5. EXPERIMENTS AND RESULTS

The symplectic maximum likelihood algorithm described in section 4 is used to study the optimal feature space for diagonal-covariance Gaussian mixture HMM modeling of the TIMIT database. The phoneme set is divided to three clusters: silence, vowel-like, and consonants. We associated with each cluster a symplectic map that is trained to maximize the likelihood of the training data that correspond to the phonemes member of the cluster.

The baseline 26-feature vector consists of 12 Mel-frequency cepstrum coefficients (MFCC), energy and their deltas. In each iteration, the new feature vector is calculated using the current symplectic transformation parameters, then the maximum likelihood estimates of the HMM model parameters are calculated. Then, the maximum likelihood estimates of the symplectic map parameters are calculated using the conjugate-gradient algorithm. After the iterative algorithm converges to a set of locally optimal HMM and symplectic parameters, the training data are transformed by the symplectic map yielding the final symplectic maximum likelihood transform (SMLT) feature vector for each cluster of phonemes.

In our experiments, the 61 phonemes defined in the TIMIT database are mapped to 48 phoneme labels for each frame of speech. These 48 phonemes are collapsed to 39 phoneme for testing purposes as in [12]. A three-state left-to-right model for each triphone is trained. The number of mixtures per state was fixed to four. The parameters of the recognizer and the symplectic map are trained using the training portion of the TIMIT database. The parameters of the triphone models are then tied together using the same approach as in [13].

To compare the performance of the proposed algorithm with other approaches, we generated acoustic features using independent component analysis (ICA), and maximum likelihood linear transform (MLLT). We tested both approaches using the same three-cluster categorization of the phoneme set used with SMLT. A cluster-dependent feature vector is designed for each cluster using maximum likelihood ICA and MMLT. We used the linear ICA approach described in [8], and implemented MLLT as described in [7].

Testing this recognizer, using the test data in the TIMIT database, we get the phoneme recognition results in table 1. These results are obtained by using a bigram phoneme language model and by keeping the insertion error around 10%. The table compares SMLT recognition results to the ones obtained by MFCC, ICA, and MLLT.

Table 1: Phoneme Recognition Accuracy

Acoustic Features	Recognition Accuracy
MFCC	73.7%
ICA	73.9%
MLLT	74.5%
SMLT	75.5%

6. References

[1] P. M. Baggenstos, "Class-specific features in classification" *IEEE Trans. On Signal Processing*, Vol. 47, pp. 3428-3432, December 1999.

[2] Andrew K. Halberstadt, *Heterogeneous Acoustic Measurements and Multiple Classifiers for Speech Recognition*, Ph.D. thesis, MIT, ECE Department, 1998.

[3] James Glass, Jane Chang, and Michael McCandless, "A Probabilistic Framework For Feature-Based Speech Recognition," *Proc. of Int. Conf. of Spoken Language Processing*, Philadelphia, PA, pp. 2277-2280, 1996.

[4] Mark J. F. Gales, "Maximum Likelihood Multiple Subspace Projections For Hidden Markov Models" *IEEE Trans. On Speech And Audio Processing*, Vol. 10, No. 2,pp. 37-47, February 2002.

[5] Alex Bailey, *Class-dependent features and multicategory classification*, Ph.D. thesis, University of South Hampton, ECE Department, 2001.

[6] R. Neal and G. Hinton, "A View of the EM Algorithm that Justifies Incremental, Sparse, and other Variants," *Learning in Graphical Models*, Kluwer Academics, 1998.

[7] R. A. Gopinath, "Maximum Likelihood Modeling With Gaussian Distributions For Classification," *IEEE Proceedings of ICASSP*, Seattle, Washington, 1998.

[8] Jong-Hwan Lee, Ho-Young Jung, and Te-Won Lee, "Speech Feature Extraction Using Independent Component Analysis," *IEEE Proceedings of ICASSP*,Vol. 3, pp. 1631-1634, Istanbul, Turkey, 2000.

[9] Thomas M. Cover, and Joy A. Thomas, *Elements of Information Theory*, Wiley, New York, NY, 1997.

[10] Athanasios Papoulis, *Probability, Random Variables, and Stochastic Processes*, McGraw-Hill, New York, 1991.

[11] L. C. Parra, *Symplectic nonlinear component analysis*, In Advances in Neural Information Processing Systems, 8, MIT Press, Cambridge, MA., pp. 437-443, 1996.

[12] Kai-Fu Lee, and Hsiao-Wuen Hon, "Speaker Independent Phone Recognition Using Hidden Markov Models," *IEEE Trans. on ASSP*, 37(11), pp. 1641-1648, November 1989.

[13] S. Young, and P. Woodland, "State Clustering in hidden Markov model continuous speech recognition," *Computer, Speech, and Language*, 8(4), pp. 369-383, October 1994.