A CONTINUOUS DENSITY NEURAL TREE NETWORK WORD SPOTTING SYSTEM *

Stephen V. Kosonocky**, Richard J. Mammone

Rutgers University Center for Computer Aids for Industrial Productivity, Piscataway, NJ 08855-1390 USA

ABSTRACT

A new classifier is described that combines the discriminatory ability of the neural tree network (NTN) with the Gaussian mixture model to create a continuous density neural tree network (CDNTN). The CDNTN is used within a Hidden Markov Model (HMM), along with a nonparametric state duration model to construct a continuous word spotting system for real time applications. The new word spotting system does not use a general background model, allowing construction of independent models whose performance is independent of the number of models in the recognition system, supporting a direct parallel implementation. Although HMM word spotting systems are shown to provide good performance when sufficient training data is available, for applications where background speech data is not available or only a limited numbers of training tokens are available, the CDNTN word spotting system is shown to out perform comparable HMM systems.

1. INTRODUCTION

Word spotting is the process of identifying the occurrence of keywords from a continuous speech utterance. Recently, a number of papers have described new keyword spotting systems. This paper presents a system based on a new continuous density neural tree network (CDNTN) and a Markov model with nonparametric state durations.

Recently attention has focused on constructing hidden Markov model (HMM) based speaker independent word spotting systems using either subword models or whole word models [2][8][9][13]. Typically a system is trained using speech data that contains keyword and non keyword tokens from a training data base, to construct a limited vocabulary recognition network consisting of keywords and a non keyword background model. Recognition can be performed using the Viterbi algorithm with the HMM network consisting of each keyword in parallel with a general background model. Performance of these systems is dependent on the number of keyword models in the system and the ability of the background model to absorb the non keyword utterances.

*This work was sponsored by Rome Laboratory, Contract No. F30602-91-C-0120.

**Presently with IBM T.J. Watson Research Center, Yorktown Heights, NY 10598

A number of speech recognition systems have used the added discriminative ability of neural networks within the state models of an HMM speech recognizer [6][15], to increase performance. This paper will introduce a new classifier, called the continuous density neural tree network (CDNTN), which combines the discriminatory ability of a neural tree network (NTN) with a mixture of gaussians. The CDNTN is used to model the posterior probability of a feature vector from a subword segment within a keyword, the subword models can be connected together to form a Markov chain to form a word model. Once a Markov chain is created for each keyword, state durations are extracted from the training data and clustered to form a state duration template to nonparametrically model the durations of each state. For testing, a dynamic time warping (DTW) algorithm is used to evaluate the state outputs for the test utterance against the state duration model extracted from the training data. The state duration template provides a temporal model for the state outputs during a keyword occurrence distinguishing between random state outputs and temporally aligned outputs during a keyword occurrence obviating the need for a recognition network.

2. Neural Tree Network State Models

A NTN is a hierarchical classifier [10] that uses a tree architecture and logistic regression to implement a sequential decision strategy based on discriminative training. Each level of the NTN uses a logistic regression algorithm (perceptron model) to divide the input training vectors into the best possible subsets according to an L_1 or L_2 cost function. Each subset is subsequently passed to a child node, and the algorithm recursively proceeds until the subset consists of a homogeneous set of classes or until some pruning criteria is satisfied [1][10]. The terminal "leaf" nodes are assigned a label corresponding to the majority class of the subset of training vectors reaching that node. During testing, an unknown feature vector is directed through the tree by the outcome of a logistic function using the weights stored in each node until a leaf node is reached. The vector is then classified according to the label at that leaf.

2.1 NTN Nonparametric Probability Estimation

Nonparametric discrete probability estimation is possible using a method such as Parzen windows [3], which samples the feature space by hypercubes defining discrete

0-7803-2431-5/95 $4.00

regions. The posterior probability of class i for a feature vector x, falling within region j can be approximated by,

$$P(C_i|x_j) = \frac{k_{ij}}{\sum_{l=1}^{M} k_{lj}} \quad (1)$$

where k_{ij} is the number of samples of class i in region j and the denominator term corresponds to the total number of samples of all classes in region j.

2.1.1 Discrete Probability Estimation using the NTN

Sampling of the feature space can also done by an NTN where each leaf defines a discrete region j in eq. (1) [4]. A pruned NTN can be used to non-uniformly sample the feature space, allowing multiple classes to be represented within each leaf region. Since the NTN uses the same cost function described in [7] to train a simple neural network within each node, the hyperplane splits can be shown to successively minimize the error of posterior probability estimation by the logistic functions. The net effect is to partition the feature space so that confused regions where the class distributions highly overlap, are sampled at a high rate while, broad homogenous regions within the feature space are coarsely sampled. Once an NTN is grown, eq. (1) can be used to provide a discrete probability of a class occurring given a training vector directed to the region j [4].

2.1.2 Continuous Probability Estimation using the CDNTN

Discrete posterior probability estimation using a window technique requires that enough samples of each class are present in the window to obtain accurate sample counts which reflect the distribution of vectors within the region. Creating windows too small using a finite number of training samples can produce binary probability estimates of 0 or 1 when the regions are homogenous, corresponding to a fully grown tree. To avoid this, a new classifier was created blending parametric and nonparametric estimation techniques similar to the graphical location models described in [14] to allow finer resolution for more accurate probability estimation. The posterior probability of class C_i defined within a discrete leaf region j can be given by Bayes' rule as,

$$P_j(C_i|x) = \frac{P_j(C_i)\,P_j(x|C_i)}{\sum_{n=1}^{M} P_j(C_n)\,P_j(x|C_n)}, \quad (2)$$

where $P_j(C_i)$ is the prior probability of class i within region j, The distribution for each class can be parameterized as a mixture of Gaussians by,

$$P_j(x|C_i) = \sum_{m=1}^{M} c_{im} N(x, \mu_{im}, \Sigma_{im}), \quad (3)$$

$$N(x, \mu, \Sigma) = \left((2\pi)^{-d/2}|\Sigma|^{-\frac{1}{2}}\right) exp\left[-\tfrac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right], \quad (4)$$

where c_{im}, μ_{im} and Σ_{im} are the weight, mean and variance, respectively, of the m^{th} mixture for class i, M is the number of classes and d is the dimension of the feature space. Since the leaf regions are not guaranteed to be statistically independent, the global posterior probability can be found by multiplying the probability of the leaf region with the local probability, given by,

$$P(C_i|x) = P(C_i|x_j)\,P_j(C_i|x). \quad (5)$$

The NTN portion of the model, partitions the feature space into distinct regions to allow a local parametric model to be developed on the data in that leaf region. For applications to subword modeling, this is depicted in Fig. 1, where a CDNTN is trained to predict the posterior probability of a subword occurring. Each substate model is a locally derived model for a region weighed by the confusability of the subword with feature vectors labeled as not belonging to the subword. During recognition, an unknown feature vector is applied, and the tree determines the appropriate substate model. The CDNTN allows locally appropriate models to be developed for classes that are inherently drawn from different distributions, i.e. different vocalizations of the same subword.

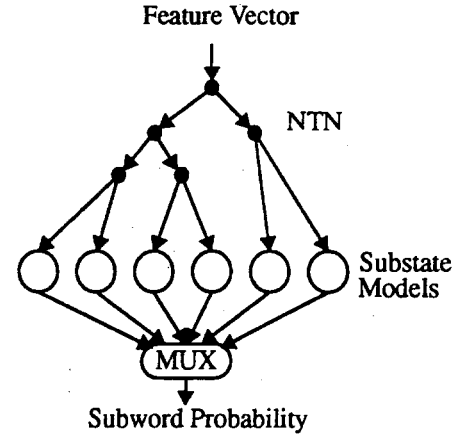

Fig. 1 CDNTN Subword State Model

3. NTN Based Word Spotting System

A word spotting system was constructed using the CDNTN as a posterior probability estimator for each state of a left-right Markov chain without skips. A forced alignment algorithm was used to create the phonetic separations within each keyword. Once the keywords were phonetically segmented, a binary CDNTN model was grown for each phonetic segment within each keyword using only the phonemes within the same keyword as anti-class vectors for

discriminant training. Once the models were trained, each speech keyword utterance from the training data base was applied to the chain of phoneme models corresponding to the keyword model, and a set of durational models was created for each keyword. The durational models were then clustered using a K-means algorithm described in [12] to obtain a master state duration template, illustrated in Fig. 2 for a sample keyword.

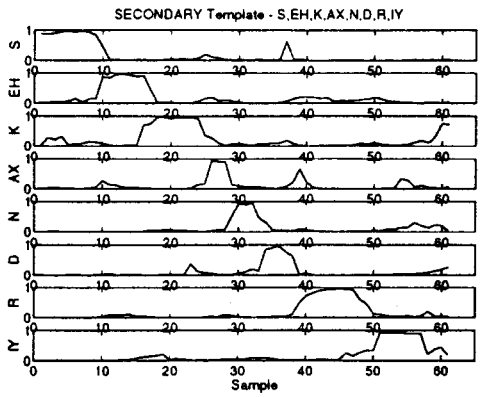

Fig. 2 Example of state duration template.

The recognition process, depicted in Fig. 3, scans each test utterance using a sliding window based on the keyword state duration template length. For each window position, the speech is parameterized into feature vectors and each vector is applied to the state models within each keyword model. The state outputs for the test utterance are then compared to the keyword state duration template using a dynamic time warping procedure. The distance score obtained from the DTW algorithm is compared against a threshold to determine a putative keyword occurrence. Fig. 4 shows the output of the system as a function of the feature sample. Training each phoneme model independently with only the speech utterances from the individual keyword tokens, allows the recognition performance to be independent to the number of keywords in the system. This method allows for a simple parallel implementation where each keyword is searched independently on a set of processors, producing a continuous stream of scores given continuous speech input for real time monitoring.

3.2 Experimental Results

Evaluation of the new word spotting system was performed using the NIST Stonehenge database [5] consisting of marked speech files for 20 keywords. Training was done using read paragraphs over actual telephone lines for 28 male and 28 female speakers from the Waterloo extension. The test set consisted of conversational speech, bandpass filtered to simulate telephone quality for 10 male speakers; sm33c through sm41c, and sm43c. The features used for the experiment consisted of the first 8 cepstral coefficients derived from 18 mel-scale filter bank outputs with an added normalized log energy term. The first and second differences were appended to create a 27 dimensional feature vector.

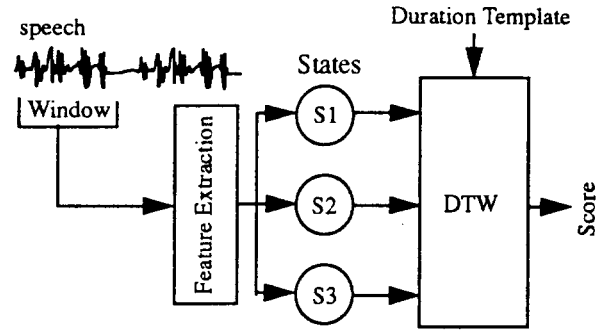

Fig. 3 Word spotting recognition system

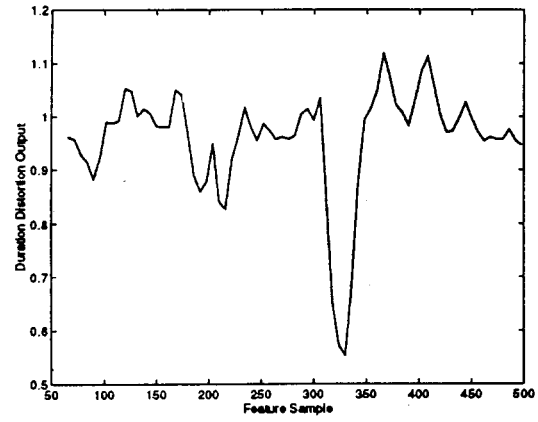

Fig. 4 Sample output distortion of duration template match for *springfield*

3.2.1 System Performance

Word spotting performance is typically judged by the detection rate (number of correctly spotted keywords/number of actual keywords) versus false alarm (FA) rate (number of insertions/keyword/hour). A figure of merit (FOM) is defined as the average detection rate for 0 to 10 FA/kw/hr. Table 1. shows the performance for each keyword, the overall FOM of 41.70% was measured, with 45.8% at 10FA/hr.

A similar test was done using an HMM word spotting network similar to [2][9][11]. Word spotting results on the same test set using a two sets of triphone models for the network, one using pooled keyword data to construct the keyword triphones, another using pooled background data for the non keyword triphones, gave an overall FOM of 58.8%. When the background model was reduced to just a silence model, the overall FOM dropped to 19.73%. When no background model was used the FOM dropped to 3.88%. Fig. 5 shows a comparison of the CDNTN system performance to two HMM systems as a function of average

number of training tokens used per keyword for all 20 keywords. The first HMM system shown with the dashed lines uses keyword pooled tokens to construct the keyword models, and pooled non keywords and silence tokens to construct the background model. The second HMM system, dot-dashed line, uses pooled keyword tokens to construct both the keyword models and the background models with an additional silence model. The CDNTN system uses only keyword tokens, and no explicit background model.

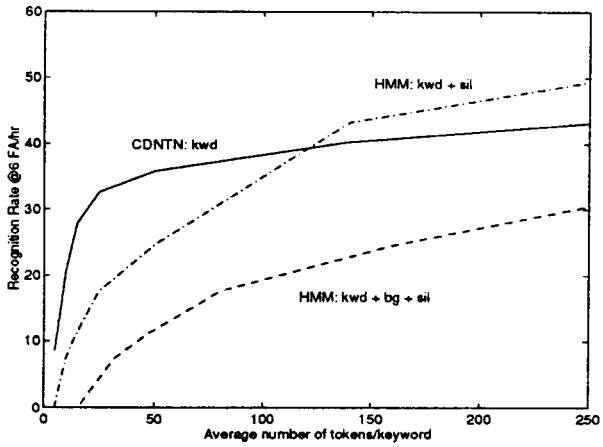

Fig. 5 Comparison of CDNTN to HMM word spotting system as a function of training tokens on male test

Keyword	FOM	Keyword	FOM
BOONSBORO	45.94	ROAD	8.63
CHESTER	28.69	SECONDARY	92.31
CONWAY	52.44	SHEFFIELD	66.67
INTERSTATE	34.67	SPRINGFIELD	75.47
LOOK	0.00	THICKET	55.39
MIDDLETON	59.09	TRACK	21.03
MINUS	46.99	WANT	8.55
MOUNTAIN	12.89	WATERLOO	51.94
PRIMARY	52.45	WESTCHESTER	69.23
RETRACE	89.75	BACKTRACK	76.97

Table 1. Performance scores for each keyword for 10 male speakers from the Stonehenge test corpus.

4. Concluding Remarks

A new classifier was described blending the discriminative nonparametric modeling capabilities of the NTN with a local parametric model based on a mixture of Gaussians. A word spotting system was developed using this classifier which accommodates parallel implementation, with performance independent of the vocabulary size and background model. Comparative experiments using HMM networks showed that it is possible to achieve superior performance using a network of vocabulary words. Experiments showed that the HMM performance degrades greatly when the background model is omitted, or when the number of training tokens is reduced. The new NTN based word spotting system described provides better performance for applications when speech data for a general background model is not available.

5. References

[1] Breiman L, Friedman J.H, Olshen R.A, Stone C.J, Classification and Regression Trees, Wadsworth International Group, Belmont, CA. 1984.

[2] Carlson B, Chang E, Lippman R, Reynolds D, Zissman M. "Wordspotting on an HTK Foundation", Speech Research Symposium XIII 1993, June.

[3] Duda R.O, Hart P.E, Pattern Classification and Scene Analysis John Wiley & Sons, New York, 1973.

[4] Farrell K, Mammone R, Assaleh K, "Speaker Recognition Using Neural Networks and Conventional Classifiers", IEEE Trans. on Speech and Audio Processing, Vol 2, No. 1, Part II, Jan. 1994.

[5] Fisher W.M, Doddington G.R, Goudie-Marhsal K.M, "The DARPA speech recognition research database: Specifications and status. In Proceedings of the DARPA Speech Recognition Workshop, pp. 93-99, 1986.

[6] Renals S, Morgan N, Bourland H, Cohen M, Franco H, "Connectionis Probability Estimators in HMM Speech Recognition", IEEE Trans. on Speech and Audio Processing, Vol. 2, No. 1, Part II, Jan. 1994.

[7] Richard M.D, Lippman R.P, "Neural Network Classifiers Estimate Bayesian a posteriori Probabilities", Neural Computation 3, pp. 461-483, 1991.

[8] Rohlicek J.R, Jeanrenaud P, Ng K, Gish H, Musicus B, Siu M, "Phonetic Training and Language Modeling for Word Spotting", Intr. Conf. on Acoustics Speech and Signal Processing, vol II, pp. 459-462, 1993.

[9] Rose R.C, Paul D.B, "A Hidden Markov Model Based Keyword Recognition System", Intr. Conf. on Acoustics Speech and Signal Processing, vol I, pp. 129-132, 1990.

[10] Sankar A, Mammone R.J, "Growing and pruning neural tree networks", IEEE Trans. on Computers, C-42, pp. 221-229, March 1993.

[11] Vrooman L.C, Narmandin Y, "Robust Speaker-Independent Hidden Markov Model Based Word Spotter", Speech Recognition and Understanding, Recent Advances, Edited by, P. Laface and R. DeMori, Springer-Verlag, Berlin, 1992.

[12] Wilpon J.G, Rabiner L.R, "A Modified K-Means Clustering Algorithm for Use in Isolated Word Recognition", IEEE Trans. on Acoustics, Speech, and Signal Processing, Vol. ASSP-33, June, 1985.

[13] Wilpon J.G, Rabiner L.R, Lee C-H, Goldman E.R, "Automatic Recognition of Keywords in Unconstrained Speech Using Hidden Markov Models", IEEE Trans. on Acoustics, Speech and Signal Processing, Vol. 38. No. 11, Nov. 1990.

[14] Whittaker J., Graphical Models in Applied Mathematical Multivariate Statistics, John Wiley & Sons, Chichester, 1990.

[15] Zeppenfeld T, Waibel A.H, "A Hybrid Neural Network, Dynamic Programming Word Spotter", International Conf. on Acoustics, Speech and Signal Proc., Vol II. pp. 77-80, 1992.