A Minimum Classification Error Based Distance Measure for Template Based Speech Recognition

Mike Matton[1], *Dirk Van Compernolle*[2], *Ronald Cools*[1]

[1]Katholieke Universiteit Leuven, Department of Computer Science
Celestijnenlaan 200A, B-3001 Heverlee (Leuven), Belgium
[2]Katholieke Universiteit Leuven, Department of Electrical Engineering
Kasteelpark Arenberg 10, B-3001 Heverlee (Leuven), Belgium
mike.matton@cs.kuleuven.be, dirk.vancompernolle@esat.kuleuven.be,
ronald.cools@cs.kuleuven.be

Abstract

In this paper we investigate the minimum classification error (MCE) criterion for the training of distance measures for template based speech recognition. These MCE-based distance measures are illustrated with example experiments on the Wall Street Journal 5k benchmark for continuous speech recognition.
Index Terms: distance measures, template based speech recognition, minimum classification error, discriminative training

1. Introduction

In previous work, we presented a discriminative distance measure on the frame-level[1], which is clearly suboptimal for speech recognition. A second distance measure was constructed using the maximum mutual information criterion, producing moderate results[2].

The minimum classification error criterion has been successfully used for the discriminative training of HMM-models for speech recognition. In our template based recognizer, HMM-models are absent and are replaced by the pure speech data (template based speech recognition). The MCE-criterion will be transformed to be usable in this template based context, using phonemes a the basic unit. We illustrate the usability of this criterion with some example experiments on the Wall Street Journal 5k benchmark.

This paper is organized as follows. In section 2 we give a brief overview of the template based speech recognizer (TBSR). Section 3 introduces the reader to the MCE-criterion, and shows how the MCE-criterion is combined with our template based approach in order to create a new training procedure for the acoustic distance measures in template based speech recognition. Next, in section 4, we present the results of some experiments on template based speech recognition. The paper ends with a conclusion summing up the most important insights discovered in this work and some pointers for future work.

2. Template based speech recognition

2.1. Motivation

Conventional speech recognition uses HMMs for modeling the acoustic speech data. These HMM models have proven to be very useful for continuous speech recognition. However, they also have some drawbacks. One of these drawbacks is the conditional independence assumption, which is clearly wrong in the case of speech recognition.

These drawbacks were the main motivation for the template based approach. In this template based approach, the speech database is segmented into basic units (templates). Recognition of new input speech occurs by comparing the input speech with the reference templates in the database (using a dynamic time warping algorithm (DTW)). The template based speech recognizer was built by De Wachter et al. [3]

2.2. System architecture

First we briefly summarize the architecture of the TBSR. Consider a database containing reference templates (phonemes). When a new piece of input speech is presented to the recognizer and feature extraction has occurred, the input speech is processed through a bottom-up template selection. This is done by computing the k-nearest neighbours for each frame of the input with the frames in the database and applying a time filter algorithm[3] to obtain a set of reference templates similar to the input. These "activated" templates are inserted into a phoneme graph. Decoding occurs by processing the input speech through this phoneme graph by a token passing algorithm, also taking into account language model scores. During the processing, several extra costs are introduced (gender mismatch, speaker change, context mismatch, etc...). The computation of the scores between templates occurs using DTW. A schematic overview is shown in fig. 1

2.3. Dynamic Time Warping

DTW can be defined as follows: the distance $\mathbf{D_{dtw}}$ between two parts of speech: an input X_1^{t-1} and a reference $Y_1^{p(t)}$ equals:

$$\mathbf{D_{dtw}}(\mathbf{X}_1^t; \mathbf{Y}_1^{p(t)}) = \min \left\{ \begin{array}{l} \mathbf{D_{dtw}}(X_1^{t-1}; Y_1^{p(t)}) + d(\mathbf{x}_t; \mathbf{y}_{p(t)})\gamma(0) \\ \mathbf{D_{dtw}}(X_1^{t-1}; Y_1^{p(t)-1}) + d(\mathbf{x}_t; \mathbf{y}_{p(t)})\gamma(1) \\ \mathbf{D_{dtw}}(X_1^{t-1}; Y_1^{p(t)-2}) + d(\mathbf{x}_t; \mathbf{y}_{p(t)})\gamma(2) \end{array} \right\}. \quad (1)$$

p is called the warping function, $\gamma(0)$, $\gamma(1)$ and $\gamma(2)$ are called the warping factors, $d(\mathbf{x}_t; \mathbf{y}_{p(t)})$ is the frame distance (or "local" distance) between frames $\mathbf{x}_t$ and $\mathbf{y}_{p(t)}$. This is discussed in the next section.

This work is sponsored by the FWO project G.0260.07 TELEX: Combining acoustic TEmplates and LEXical modeling for improved speech recognition

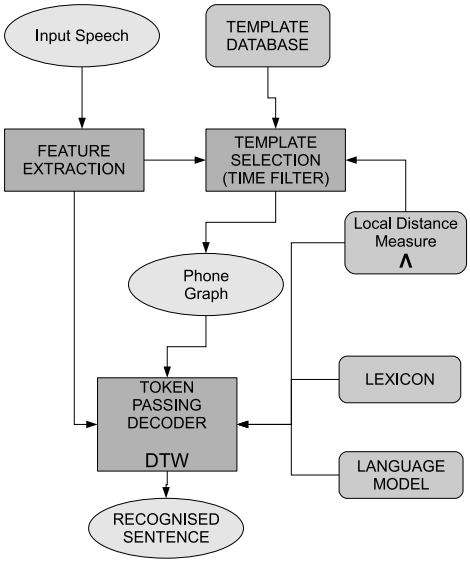

Figure 1: System Architecture

2.4. Local Distance Measure

The most simple solution for this local distance would be the Euclidean distance between the two frames. However, previous research has shown that scaling this Euclidean distance measure is suboptimal [4, 1]. We will discuss some alternatives.

2.4.1. Diagonal Weight Matrix

A second possibility is to divide the speech frames into classes and to scale each class separately. In this case, the local distance measure becomes

$$d(\mathbf{x};\mathbf{y}) = \sum_{j=1}^{D} \lambda_{j,l} \left(x_j - y_j\right)^2, \mathbf{x},\mathbf{y} \in \mathbb{R}^D, \qquad (2)$$

where l is the class to which $\mathbf{y}$ belongs. The set of weights $\lambda_{j,l}$ is described by Λ throughout this paper.

This basic weighting of the distance measure uses D parameters per class, where D is the dimension of the feature space (diagonal weight matrix). This is only correct if the points in the different classes are normally distributed along the basic axes of the search space. In practice this is not the case. It is easy to show that correllations between features do appear.

2.4.2. Full Weight Matrix

A first solution for this problem would be to use a full weight matrix per class, i.e.,

$$d(\mathbf{x};\mathbf{y}) = (\mathbf{x}-\mathbf{y})^t \Lambda_l (\mathbf{x}-\mathbf{y}). \qquad (3)$$

However, this solution makes the search space for the weights too large. The recognizer currently uses 25-dimensional feature vectors and about 1500 different classes, leading to a 937500-dimensional search space. Moreover, some classes have too few examples available to train a full weight matrix. Last but not least, also the decoding would be very time consuming in this case.

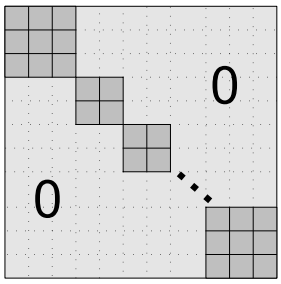

Figure 2: Block diagonal weight matrix

2.4.3. Block Diagonal Weight Matrix

To overcome this, the number of parameters in the weight matrix should be reduced. We have done this by constructing a sparse weight matrix. We created a "block diagonal" matrix in which only the most important dependencies between two features (in the feature vector) are included. These dependencies can be checked by computing the correlations between every pair of features in the feature vector. This has been done for each class separately so the form of the weight matrices is different for every class. In this way we created matrices with local "blocks" of features that are correlated. These blocks consist of at most 3 features. It is clear that using this approach leads to weight matrices with at most $3n$ parameters. Note however that the resulting matrix is not really block diagonal. I.e. the matrix can be transformed into a block diagonal matrix with a set of appropriate row and column permutations. In fig. 2, such a weight matrix is shown after the necessary row and column permutations. Note that this new approach has a computational advantage. Calculations with sparse weight matrices are a lot less time consuming than full matrix computations. The Sparse BLAS routines are used to perform the computations[5].

3. Minimum Classification Error

Several discriminative approaches for pattern classification have been proposed over the years. The two most commonly used criteria are maximum mutual information (MMI) and minimum classification error (MCE)[6].

When performing a discriminative training approach, the definition of the classes needs careful consideration. There are several options for choosing the basic unit: we could use phonemes, words, or even sentences. For this work we have chosen the phoneme as the basic unit. Phonemes are also the basic unit in the template based speech recognizer used for the experiments. This basic unit leads to a very straightforward definition of the classes, namely the (context independent) phoneme. Using context dependent phonemes is much more troublesome. In template based recognition, a phoneme with a wrong context is not necessarily a bad one, so it is not necessary to penalise it.

We would also like to point out that these basic units for MCE training are significantly different from the classes used for the local scaled distance measure. These classes can be allocated on a frame by frame basis. As explained in section 4.1, these classes correspond to context-dependent states derived from a phonetic decision tree in our current setup.

3.1. The MCE criterion

Suppose we have a set of training examples (which will be templates in our recognizer) and this set is divided into different classes $C_k, k = 1..K$, and suppose we have a new input vector $\mathbf{x}$. This input vector will be classified in class C_k if

$$g_k(\mathbf{x}, \Lambda) = \max_{i \in K} g_i(\mathbf{x}, \Lambda), \tag{4}$$

where $g_i(\mathbf{x}, \Lambda)$ represents a discriminative function that indicated how well $\mathbf{x}$ fits in class C_i. The larger the difference between the scores for the correct class and the competing wrong classes, the better classification will work. This observation naturally leads to a criterion in which the classification error is minimized:

$$d_k(\mathbf{x}, \Lambda) = -g_k(\mathbf{x}, \Lambda) + \left(\frac{1}{N-1} \sum_{i \neq k} (g_i(\mathbf{x}, \Lambda))^\eta \right)^{1/\eta}. \tag{5}$$

In the extreme case that η approaches infinity, this equation degrades to

$$d_k(\mathbf{x}, \Lambda) = -g_k(\mathbf{x}, \Lambda) + g_i(\mathbf{x}, \Lambda), \tag{6}$$

with C_i the best competing class for input $\mathbf{x}$.

Next, this criterion is passed through a continuous loss function, which typically has values between 0 (a misclassification) and 1 (a correct classification). One possible loss-function is a sigmoid. The resulting loss is then defined as follows:

$$l_k(\mathbf{x}, \Lambda) = f(d_k(\mathbf{x}, \Lambda)) = \frac{1}{1 + e^{-\xi d_k(\mathbf{x}, \Lambda)}}. \tag{7}$$

Minimizing this criterion results in minimizing the classification error. In the ideal case the loss is 0, meaning that the input vector has been correctly classified without doubt.

3.2. MCE training on the template database

As said in the introduction of this section we have chosen context-independent phonemes with gender information as the basic unit for MCE training.

First, we have to choose the measure of similarity between a new acoustic observation $\mathbf{A}$ and a class C_k: $g_k(\mathbf{A}, \Lambda)$. Since the similarity of speech utterances is computed using DTW in TBSR, a straightforward measure would be the mean DTW distance between $\mathbf{A}$ and its nearest neighbours in class C_k. This results in the following equation:

$$g_k(\mathbf{A}, \Lambda) = \sum_{k\text{NN } \mathbf{X_m} \text{ of } \mathbf{A} \text{ in } C_k} \mathbf{D_{DTW}}(\mathbf{A}, \mathbf{X_m^A}). \tag{8}$$

In this case, Λ represents the set of weights for the locally weighted distance measure defined in equation (2). We have chosen the k nearest neighbours (kNN) to reduce the influence of outliers in the training and to catch the optimal direction for training in a better way. k is chosen beforehand and thus is a model parameter.

When we insert this into the MCE criterion, using the assumption that $\eta \to \infty$ we obtain

$$d_k(\mathbf{A}, \Lambda) = - \sum_{\text{NN } \mathbf{X_m} \text{ of } \mathbf{A} \text{ in } C_k} \mathbf{D_{DTW}}(\mathbf{A}; \mathbf{X_m^A}) + \sum_{\text{NN } \mathbf{X_m} \text{ of } \mathbf{A} \text{ in } C_i} \mathbf{D_{DTW}}(\mathbf{A}; \mathbf{X_m^A}). \tag{9}$$

```
1  mce_train() {
2    Lambda := init_distance;
3    old_mce_test :=  compute_mce(test_set, Lambda);
4    repeat {
5      DLAMBDA = 0;
6      foreach speaker s{
7        foreach i,j
           { d_lambda(i,j) = 0; }
8        foreach template not from s {
9          g := compute_gradient(
                       $1_k(A, Lambda));
10          foreach weight lambda(i,j) {
11           d_lambda(i,j) += g(i,j);
12         }
13       }
14       DLAMBDA += sum(d_lambda(i,j));
15       lambda(i,j) += d_lambda(i,j);
16       new_mce_test :=
           compute_mce(test_set, Lambda);
17       if new_mce_test > old_mce_test
18         end training;
         old_mce_test := new_mce_test;
19     }
20   }
21   until DLAMBDA < threshold
22 }
```

Figure 3: *Pseudo-code of simplified MCE training algorithm.*

To this criterion we apply the sigmoid loss function, resulting in the misclassification measure $l_k(\mathbf{A}, \Lambda)$. If we then sum this loss function over all classes and training examples, the total loss function $L(\Lambda)$ is obtained:

$$L(\Lambda) = \sum_{k=1}^{K} \sum_{i=1}^{N_k} \frac{1}{1 + e^{-\xi d_k(\mathbf{A}_i, \Lambda)}}. \tag{10}$$

In this equation, K is the number of classes, and N_k equals the number of available templates for class k.

In order to minimize the classification error, we have to minimize the total loss function. This is an optimization problem for a continuous function. We have chosen a gradient descent approach for minimizing the function, i.e., we have to compute the following derivative:

$$\frac{\partial L(\Lambda)}{\partial \lambda_{v,w}} = \sum_{k=1}^{K} \sum_{l=1}^{N_k} \frac{\partial}{\partial \lambda_{v,w}} \frac{1}{1 + e^{-\xi d_k(\mathbf{A}_{k,l}, \Lambda)}}, \tag{11}$$

$$= \sum_{k=1}^{K} \sum_{l=1}^{N_k} \frac{\xi e^{-\xi d_k(\mathbf{A}_i, \Lambda)}}{(1 + e^{-\xi d_k(\mathbf{A}_i, \Lambda)})^2} \frac{\partial}{\partial \lambda_{v,w}} d_k(\mathbf{A}_{k,l}, \Lambda)$$

in which $d_k(\mathbf{A}_i, \Lambda)$ is obtained from eq. (9) Note that eq. (9) is a sum of DTW-distances and that a DTW distance equals the sum of local distances between speech frames. Therefore, the derivative of $d_k(\mathbf{A}_{k,l}, \Lambda)$ with respect to $\lambda_{v,w}$ is computed by taking the sum of the local distances in which w is the class of the speech frame that the input is compared to for that particular frame. Which local distances were computed in order to obtain the DTW distance between two templates can easily be checked by storing back-pointer information in the DTW matrix. If this procedure is followed for every template in the training database, an update for every $\lambda_{v,w}$ in the direction of the gradient is obtained after one iteration. The weights of the distance measure are updated after each iteration and the procedure starts over until convergence occurs.

The main structure of the training algorithm is shown in fig. 3.

nov92	Eucl	Loc Mah	MCE 1	MCE 2
NDS	9.8	9.4	9.3	8.9

Table 1: Word error rates for the MCE criterion.

4. Experiments

4.1. Experimental set-up

All experiments are performed using the template based speech recognition software, constructed by De Wachter et al.[3]. The reference database for the experiments is the Wall Street Journal 5k benchmark[7] using the nov92 test set.

The classes for the local weighted distance measure are based on the CD-HMM state numbers obtained using a forced alignment based on phonetic decision trees of the WSJ training data, together with the gender information of the speaker. Preprocessing is done using "mida" features[8], leading to 25-dimensional feature vectors. The basic unit used in the recogniser is a phoneme. These phonemes also define the basic unit for the MCE discriminative training algorithm. We have used 5 nearest neighbours for each template to compute the measure of similarity. This parameter was chosen experimentally.

In order to train the MCE criterion, the WSJ5k database is split up into a training, a validation and a test set. The computation of misclassification measures is done using a leave-one-speaker-out approach. Training ended when the MCE measure for the validation set no longer improved (indicating overtraining).

4.2. Experimental results

Experimental results with the MCE distance measure on the nov92 test set are listed in table 1. The word error rate is shown for recognition using the baseline Euclidean distance (column 2) and for the MCE-based distance measure in two different variants (column 4 and 5). The first variant uses the weighted local distance measure with diagonal weight matrix (MCE1). The second one uses the sparse weight matrix discussed in section 2.4 (MCE2). It is also compared with a local mahalanobis distance measure proposed by De Wachter et al. [4] (column 3).

We can observe that using the MCE-based distance measure improves the baseline Euclidian distance. The improvement of the MCE 1 measure is not significant, but the MCE 2 distance measure improves the accuracy with 9.3% relatively. The local (diagonal) Mahalanobis distance measure is improved by 5.3% relatively.

For the reference: recent work by Fu et al. on MCE and HMM-based speech recognition using HTK shows a WER of 7.96% and an MLE-baseline of 8.41% on WSJ0[9].

5. Conclusion

In this paper, we have explained the criterion of minimum classification error and have combined this criterion with the ESAT template based speech recogniser. Several different alternatives for creating a distance measure were investigated. We have created an elegant way to increase the number of parameters of the distance measure using sparse matrices, thus without increasing the decoding time very much.

These distance measures were evaluated on the WSJ-5k benchmark using the nov92 test set. We have shown that the MCE-based distance measure results in a relative improvement of about 9.3% if compared with the basic Euclidean one.

6. Future Work

There is still room for improvement, some seem easier than others. First, De Wachter et al. have shown that doing outlier correction can improve the results with about 10% relatively[10]. This seems a straightforward extension of this work.

A second possible improvement concerns the fact that the use of phonemes is not the ideal choice for computing the MCE criterion. Since the ultimate goal in the experiments is to minimise the word error rate, the use of a whole word as a basic unit would be a better alternative. However this poses new problems, as many words have pronunciation variants, which are difficult to model in an MCE context.

7. References

[1] M. Matton, M. De Wachter, D. Van Compernolle, and R. Cools, "A discriminative locally weighted distance measure for speaker independent template based speech recognition," in *Proceedings of ICSLP 04*, Jeju, October 2004, pp. 429–432.

[2] ——, "Maximum mutual information training of distance measures for template based speech recognition," in *International Conference on Speech and Computer*, Patras, Greece, October 2005.

[3] M. De Wachter, M. Matton, K. Demuynck, P. Wambacq, R. Cools, and D. Van Compernolle, "Template based continuous speech recognition," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 15, no. 4, pp. 1377–1390, May 2008.

[4] M. De Wachter, K. Demuynck, P. Wambacq, and D. Van Compernolle, "A locally weighted distance measure for example based speech recognition," in *Proceedings of ICASSP 04*, Montreal, Canada, May 2004, pp. 181–184.

[5] I. Duff, M. Heroux, and R. Pozo, "An overview of the sparse basic linear algebra subprograms: The new standard from the blas technical forum," *ACM Transactions on Mathematical Software*, vol. 28, no. 2, pp. 239–267, June 2002.

[6] B.-H. Juang and S. Katagiri, "Discriminative learning for minimum error classification," *IEEE Transactions on Signal Processing*, vol. 40, no. 12, pp. 3043–3054, December 1992.

[7] D. B. Paul and J. M. Baker, "The design for the wall street journal-based csr corpus," in *Proceedings of ICSLP 92*, Banff, Canada, October 1992, pp. 899–902.

[8] K. Demuynck, J. Duchateau, and D. Van Compernolle, "Optimal feature sub-space selection based on discriminant analysis," in *Proceedings of EUROSPEECH 99*, Budapest, September 1999, pp. 1311–1314.

[9] Q. Fu, A. Moreno-Daniel, and B.-H. Juang, "Generalization of the minimum classification error (mce) training based on maximizing generalized posterior probability (gpp)," in *Proceedings of Interspeech 2006*, Pittsburgh, USA, September 2006, pp. 681–684.

[10] M. De Wachter, K. Demuynck, and D. Van Compernolle, "Outlier correction for local distance measures in example based speech recognition," in *Proceedings of ICASSP 07*, Honolulu, Hawaii, April 2007, pp. 433–436.