Evaluating Acoustic Distance Measures for Template Based Recognition

Mathias De Wachter, Kris Demuynck, Patrick Wambacq and Dirk Van Compernolle

Katholieke Universiteit Leuven –Dept. ESAT
Kasteelpark Arenberg 10, B-3001 Leuven
{mathias.dewachter,kris.demuynck,patrick.wambacq,dirk.vancompernolle}@esat.kuleuven.be

10.21437/Interspeech.2007-321

Abstract

In this paper we investigate the behaviour of different acoustic distance measures for template based speech recognition in light of the combination of acoustic distances, linguistic knowledge and template concatenation fluency costs. To that end, different acoustic distance measures are compared on tasks with varying levels of fluency/linguistic constraints. We show that the adoption of those constraints invariably results in an acoustically clearly suboptimal template sequence being chosen as the winning hypothesis. There are strong implications for the design of acoustic distance measures: distance measures that are optimal for frame based classification may prove to be suboptimal for full sentence recognition. In particular, we show this is the case when comparing the Euclidean and the recently introduced adaptive kernel local Mahalanobis distance measures.

Index Terms: template based, example based, episodic, non-parametric

1. Introduction

In our template based speech recognizer, the best sentence hypothesis is based on the single best template string (Viterbi approximation). The score of the template string hypothesis is a combination of the acoustic distance between the individual templates and their respective segments of the input speech, a language model score and a *prior* probability for the template sequence (a fluency measure). Since the best scoring hypothesis is obtained by a joint optimization given all three knowledge sources, the winning hypothesis will invariably contain acoustically suboptimal templates. In this paper, we discuss the implications of this phenomenon for the design of between-frame acoustic distance measures.

In a set of experiments, we give classification/recognition results for a template based system at different levels: The first two experiments are frame and phone based *classification*, i.e. they are based on acoustic resemblance only. In the next two experiments, phone string *recognition* and sentence recognition, the dynamic time warping (DTW) scores are combined with both the prior template sequence probability model and extra lexical and language model constraints.

Section 2 gives an overview of our template based recognizer and defines the different distance measures. Section 3 describes the experiments. Section 4 discusses the experimental results and explains the apparently paradoxical behaviour of the distance measures over the different experiments.

This research was funded by the Fund for Scientific Research Flanders (FWO-project G.0260.07 "TELEX").

2. Overview of a template based recognizer

This section briefly describes the template based recognition system used in this paper. More details can be found in [1, 2].

2.1. Template based Bayesian recognition

In typical speech recognition systems, the most likely word string (sentence) $\hat{\mathbf{w}}$ is found by using a combination of the acoustic model and a language model:

$$\hat{\mathbf{w}} = \underset{\mathbf{w}}{\operatorname{argmax}}\, P(\mathbf{w}|\mathbf{x}) = \underset{\mathbf{w}}{\operatorname{argmax}}\; f(\mathbf{x}|\mathbf{w})P(\mathbf{w}) \qquad (1)$$

where $\mathbf{x}$ denotes the input sequence and $\mathbf{w}$ denotes a word sequence. $f(\mathbf{x}|\mathbf{w})$ is the acoustic likelihood of the input, and $P(\mathbf{w})$ is the language model probability. Since the derivation is simply an application of Bayes' rule,[1] this approach is often called the *Bayesian recognition paradigm*.

In a hidden Markov model (HMM) speech recognizer, the acoustic model $f(\mathbf{x}|\mathbf{w})$ consists of a single sequence of (context-dependent) HMM states that corresponds to the word string (assuming there are no pronunciation variations). By contrast, in a template (or example) based system, *many* different template sequences are available to model the word string. Formally, a template based acoustic model is obtained by

$$f(\mathbf{x}|\mathbf{w}) = \sum_{\mathbf{t}} f(\mathbf{x},\mathbf{t}|\mathbf{w}) = \sum_{\mathbf{t}} f(\mathbf{x}|\mathbf{t},\mathbf{w})P(\mathbf{t}|\mathbf{w}) \qquad (2)$$

where $\mathbf{t}$ represents a template sequence. The new term $P(\mathbf{t}|\mathbf{w})$ expresses the *prior* probability of the template string, given the word string. This term has two functions: it expresses a *fluency* of the template sequence, very similar to the use of concatenation costs in concatenative speech synthesis, and secondly it normalizes for the frequency of occurrence of the different phones in the template database.

In practice, summing over all possible template strings to find the best word sequence is computationally infeasible. Instead, we use the Viterbi approximation (cf. [3]), replacing the sum by a maximum. This leads to the following equation:

$$\hat{\mathbf{w}} = \underset{\mathbf{w}}{\operatorname{argmax}}\; \max_{\mathbf{t}} f(\mathbf{x}|\mathbf{w},\mathbf{t})P(\mathbf{t}|\mathbf{w})P(\mathbf{w}) \qquad (3)$$

Because of the Viterbi approximation, $P(\mathbf{t}|\mathbf{w})$ no longer has to normalize the frequency of occurrence of the different phones in the template database, but we still want to model the fluency of template string hypotheses. In this context, the conditioning on $\mathbf{w}$ can be reduced to a lexical lookup; we therefore drop the condition and assume template sequence probabilities are only assigned when they form a lexically correct sequence.

[1] Note that the denominator $f(\mathbf{x})$ is dropped since it is constant in the maximization.

Given the constraints of a left-to-right decoder, the prior template string probability is estimated based on a first order Markov chain:

$$P(\mathbf{T}) = P(T_1^{N_T}) = P(T_1)\prod_{i=2}^{N_T} P(T_i|T_{i-1}) \quad (4)$$

The individual *transition probabilities* $P(T_i|T_{i-1})$ are based on a set of relevant features that aim to capture fluency (or naturalness) information. In practice, we based the transition probabilities on mismatches in phonetic context and on gender mismatches. Templates that are successors in the reference database (*natural successors*) always get a transition probability of one. This corresponds to the widely used concept of *non-uniform units* in concatenative speech synthesis [4]. Since all the model scores are combined in the negative log-domain (the negative logarithm of $f(\mathbf{x}|\mathbf{t},\mathbf{w})$ corresponds to the dynamic time warping (DTW) score of the input given a template string), the template transition probabilities become transition *costs* in our recognizer. We used fixed costs for each type of mismatch, which were optimized on a development test set.

2.2. Between-frame distance measures

Distances between input vectors $\mathbf{x}$ and reference vectors $\mathbf{y}$ in a DTW setting are most often calculated using the (squared) Euclidean distance measure:

$$d_{EU}(\mathbf{x};\mathbf{y}) = (\mathbf{x}-\mathbf{y})'(\mathbf{x}-\mathbf{y}) \quad (5)$$

The use of a Euclidean distance is reasonable in our system as a global decorrelation and normalization of the features is performed in the feature extraction setup [5]. However, since there still remains a significant amount of phone class dependent variability, the Euclidean distance is expected to be suboptimal. We therefore introduced a class dependent local scaling for between-frame distances in [6], resulting in the local Mahalanobis distance:

$$d_{LM}(\mathbf{x};\mathbf{y}) = \frac{1}{2}(\mathbf{x}-\mathbf{y})'\hat{\mathbf{\Sigma}}_c^{-1}(\mathbf{x}-\mathbf{y}) + \log|\hat{\mathbf{\Sigma}}_c|^{1/2} \quad (6)$$

The class-dependent covariance matrices (c is the class of reference vector $\mathbf{y}$) are estimated on all the vectors that have received the same label in a (context-dependent) HMM state alignment.

More recently [2], we extended the local Mahalanobis distance using two techniques from non-parametric density estimation. Each reference vector is now seen as a *kernel* (in practice a Gaussian pdf) with a certain covariance estimate and a certain *bandwidth* (a scaling factor on the covariance matrix). The *adaptive kernel local Mahalanobis distance* is defined as

$$d_{AKLM}(\mathbf{x};\mathbf{y}) = \frac{1}{2\lambda_y^2}(\mathbf{x}-\mathbf{y})'\hat{\mathbf{\Sigma}}_c^{-1}(\mathbf{x}-\mathbf{y}) + \log\left(\lambda_y^{M/2}|\hat{\mathbf{\Sigma}}_c|^{1/2}\right) \quad (7)$$

with λ_y the *local bandwidth factor* for reference vector $\mathbf{y}$, determined using a k nearest neighbours (kNN) density estimate within the same class. The idea of adding the local bandwidth factor is to base each covariance estimate on the local density around the kernel. A single parameter can reliably be estimated on a small number of data points, while estimating a covariance matrix needs significantly more data.

The second technique, called *data sharpening*, replaces each reference vector with an average of the same local neighbourhood used to determine the local bandwidth:

$$\hat{\psi}(\mathbf{y}_i) = \frac{1}{k}\sum_{j=0}^{k-1}\mathbf{y}_{(j)} \quad (8)$$

with $\mathbf{y}_{(j)}$ the j^{th} nearest neighbour (in local Mahalanobis sense) of $\mathbf{y}_i$ within the same class, and $\mathbf{y}_{(0)}$ being $\mathbf{y}_i$ itself. For both extensions, we used $k = \lceil\sqrt{n_c}\rceil$, with n_c the number of kernels per class.

2.3. Bottom-up template selection

Despite the use of the Viterbi approximation, the search space of a template based decoder based on equation 3 is still prohibitively large. For the WSJ0 task used in the experiments, the reference database contains over 500k phone templates, and the phone set contains 43 symbols. Each template can be used to represent its corresponding phone, hence a straightforward implementation will result in a search space that is over 10 000 times as complex as the equivalent HMM search space.

Therefore, we use a bottom-up selection of acoustically well-matching templates [7, 1]. The bottom-up selection algorithm computes a number of nearest neighbours (around 8200 in the presented experiments) for each input frame, and uses a fast DTW-like algorithm on the sparse matrix of the smallest distances between input and reference frames to find likely templates. The output of the bottom-up template selection algorithm is a graph where each node is an input frame number, and each arc represents a template selection.

2.4. Decoding

The template based decoder searches the bottom-up activation graph for the best template sequence, while combining the DTW scores with template transition and language model scores, and taking into account lexical constraints. Since the activation graph contains gaps and overlaps, each activation can additionally be extended with its natural successor in the reference database, the end time being estimated based on the relative difference between the length of the activation and the length of the matching input segment.

3. Experiments

In a series of experiments, we investigate recognition performance of the same system on the same test data, adding more knowledge sources in each experiment. We start with frame classification experiments exclusively based on between-vector acoustic distances. Next, the constraint that frames are part of a template is introduced, resulting in phone classification. Then, in a phone string recognition experiment we add the prior template sequence probability model and a probabilistic phone transition model (a phone trigram). Finally we add the linguistic knowledge sources in a full sentence recognition experiment.

3.1. Experimental setup

All experiments are evaluated on the WSJ November 92 5k bigram non verbalized punctuation benchmark. We prefer a word recognition task over a phone recognition task (e.g. TIMIT) since this allows us to evaluate the behaviour of the acoustic distance measure over the full range of possible usage patterns, i.e. from purely acoustic based frame classification all the way up to the interaction with lexicon and language model. A disadvantage is that in order to allow phone classification and recognition tests, a phone segmentation is needed. We therefore segmented both the training database and the test set by means of a forced alignment using an existing reference HMM system.

The HMM uses 1078 cross-word context-dependent tied states which share a pool of 17 932 diagonal covariance Gaussians. 25 dimensional feature vectors were obtained by means

	No data sharpening			Data sharpening		
	EU	LM	AKLM	EU	LM	AKLM
1-best	49.8	44.2	49.5	60.7	55.1	58.4
Voting	60.0	56.0	57.7	62.8	59.1	60.3
Parzen	56.7	53.4	56.1	62.8	59.4	60.3

Table 1: Frame classification rates for the different distance measures, without and with data sharpening.

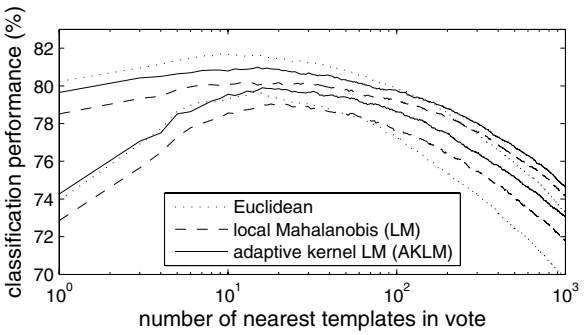

Figure 1: Phone classification performance for a kNN voting classifier for increasing k. The higher line of each type corresponds to the use of data sharpening.

of a mutual information based discriminant linear transformation (mida) on 24 MEL spectra (30ms Hamming window, 10ms frame shift) and their first and second order time derivatives [5]. We use a set of 43 phones, based on the CMU v0.6d phonetic lexicon. The SI-84 WSJ0 database, which contains 15 hours of noise-free read speech, is used for training. The training database contains contains 592 223 phone segments (including silence) and about 4.5 million speech (non silence) frames.

The template based system uses the same preprocessing, phone set and lexicon. All template experiments use diagonal covariance matrices $\hat{\mathbf{\Sigma}}_c$, where the class assignment is based on the state segmentation from the reference HMM system. For the phone string recognition and the sentence recognition, the number of bottom-up template activations is on average (almost) the same for the different distance measures.

3.2. Frame classification

The first experiment classifies each non-silence input frame based on the identity of its nearest reference vectors, according to the different distance measures. Table 1 summarizes the results for a single best classifier, a voting classifier and a Parzen based classifier. The results for the voting classifier are those for the optimal number of votes for each distance measure. The Parzen based classifier chooses the phone class for which the *average* kernel likelihood score of the input is the smallest.

3.3. Phone classification

For the phone based classification experiments, the classification is based on the DTW distance between the input segment and the reference templates. Figure 1 shows the voting classification rate for the different distance measures for an increasing number of nearest templates in the vote. The higher line of each type corresponds to the use of data sharpening.

We also investigate which reference frames are used in the closest phone template (for the Euclidean distance after data sharpening). The alignment path is retrieved from the DTW algorithm. Then, for each input frame $\mathbf{x}_\tau$, the distance to all reference frames is calculated. We define two statistics: (1) the rank order of the reference frame chosen by the DTW algorithm in the sorted set of distances, and (2) the "nearest neighbour (NN) percentile" which is the rank order divided by the number of frames in the database.

No data sharpening			Data sharpening		
Eucl.	LM	AKLM	Eucl.	LM	AKLM
21.6	18.7	18.1	14.4	14.3	14.0

Table 2: Phone string recognition error rates for the different distance measures, without and with data sharpening.

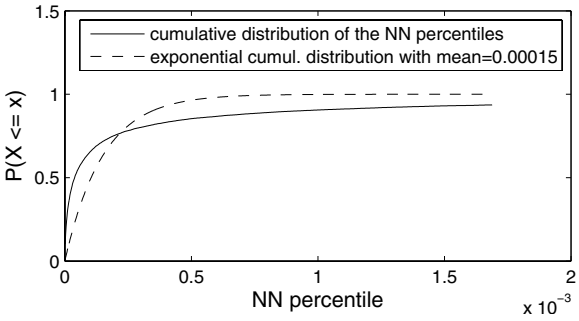

Figure 2: Cumulative distribution of the NN percentile of the templates that make up the best hypothesis in a phone string recognition experiment.

The average rank order was 1983, which corresponds to a NN percentile of 0.036%. However, most input frames are paired to a low ranking reference frame in the best matching DTW alignment. The average is heavily influenced by a few large rank numbers. The median rank, which is not influenced by these heavy tails, is 119.

Further investigation of the set of reference frames that are closer neighbours shows that some frames correspond to different phones. The median number of frames in that set corresponding to an different phone is 33. The median *phone rank* of the reference frame selected by the DTW algorithm is 3, i.e. frames from two other phones have a smaller frame rank.

3.4. Phone string recognition

In the phone string recognition experiment, complete test sentences are used as input, and the recognizer finds the most likely template strings. Apart from the template transition probabilities/costs described in 2.1, we use a trigram phone transition model, estimated on the training database.

Table 2 shows the phone error rate for these experiments. By comparison, the HMM system used to segment the training database (see section 3.1) achieved a phone error rate of 16.7% on the same task. A more complex HMM (36-dimensional features and 1818 tied states) improves that score to 14.9%.

A similar experiment as the one above checks the neighbour rank of the reference templates that are chosen during phone string recognition. From the winning phone sequence hypothesis, a phone based segmentation is obtained. For each of these segments, the nearest templates in the complete reference database are calculated. Figure 2 shows the observed cumulative distribution of the NN percentiles and, for reference, an exponential cumulative distribution with mean equal to 0.015%. The figure shows that the observed distribution is heavy-tailed. The average NN percentile for templates with a NN rank below

1000 (the maximal rank in our experimental setup) is around 0.015% (which corresponds to the 87^{th} closest template), but it can be seen that a significant percentage of the templates (6.7%) is not in the 1000-best set. The median NN percentile and rank of the chosen templates are 0.004% and 23, respectively.

Why are templates that are clearly sub-optimal acoustic matches (the 6.7% that are not in the 1000-best list) part of the best recognition hypothesis? The first possible reason is a mistake of the either the acoustic model or of the reference transcription. The second (more important) reason are contextual influences, consisting of both fluency costs and linguistic constraints.

3.5. Sentence recognition

In the final experiment, the Euclidean and adaptive kernel local Mahalanobis distance (both after data sharpening) are compared in a sentence recognition task, using the standard WSK 5k bigram language model. Using the Euclidean distance measure, the word error rate (WER) is 8.22%, while the adaptive kernel local Mahalanobis distance achieves a WER of 8.11%.

Contrary to the phone string recognition setup, *hard constraints* (the lexicon) on the template string are introduced in sentence recognition. Because of these hard constraints, a single error can lead to a series of templates that do not match the input. Therefore, the average NN rank of the chosen templates can be expected to be larger than in phone string recognition. However, the parameter optimization on the development test set assigned much smaller values to the different template transition costs (or, equivalently, a larger importance to the DTW score). As a result, the combined impact of linguistic and fluency constraints is even less than for phone string recognition. The average NN percentile in this experiment is 0.007% (again not counting the templates that are not in the 1000-best list), corresponding to on average the 42^{nd} (median 6) closest neighbour. The distribution of the neighbour ranks is similar (very heavy-tailed) to the one shown in figure 2.

4. Discussion

The experimental results show a paradoxical behaviour of the distance measures over the different tasks: the Euclidean distance clearly performs best in the classification tasks (frame and phone classification), but the scaled distances outperform the Euclidean distance in the recognition tasks (phone string and sentence recognition), although only very slightly. Without data sharpening, the difference is clearer: in that case the scaled distances easily outperform the Euclidean distance measure in the recognition tasks.[2]

The paradox can be explained by considering the variation modeled by the class-dependent covariance matrices. Within a context-dependent state, three major types of variation are present: (1) variation based on different speakers which is likely to cause multiple modes in the distribution, (2) variation caused by the within-state trajectory, and (3) 'random' variation.

When enough (relevant) data is available, there will always be reference data close to the input, and only variation of type 3 should be modeled. However, when the input is located far from the reference template, information about the first type of variation will become necessary to correctly judge the likelihood of such a large distance. The locally scaled distances take into account all three types of variation, and are therefore less suitable for short-range classification, but better equipped to correctly handle large distances (unseen input). The Euclidean distance, being class-independent, is not likely to be influenced by the variation of types (1) and (2), making it better suited for short-range classification.

In practice, adding constraints (template structure, fluency costs and linguistic knowledge) corresponds to forcing the system to use long-range acoustic distances. In this case, the local scaling is useful, as is shown in figure 1 for large numbers of templates in the vote, and by the better recognition results of the locally scaled distances.

It should be noted that modeling variation of type 2 is not helpful in template based recognition, since the DTW algorithm compares sequences with other sequences. Clustering for the covariance matrices based on DTW alignments instead of on HMM state segmentations will be investigated further.

[2] The reason for the limited improvement observed with the AKLM distance w.r.t the Euclidean distance after data sharpening on the Nov92 evaluation test set is as yet unclear to us. Both on the Resource Management task and on the WSJ development test set, the adaptive kernel local Mahalanobis distance clearly outperforms the Euclidean distance, even after data sharpening.

5. Conclusion

In this paper, we showed that in template based recognition, better frame based classification performance does not necessarily lead to better overall recognition. Specifically, we showed that local scaling performs better for phone string and sentence recognition, while the Euclidean distance was clearly better for frame and phone classification. We showed that the cause of this apparent paradox lays in the combination of the DTW scores with linguistic knowledge and (especially) the fluency constraints on the hypothesized template sequence. The locally scaled distances perform better in classifying input based on long-range distances, a feature that is useful in the phone string and sentence recognition setup.

6. References

[1] M. De Wachter, M. Matton, K. Demuynck, P. Wambacq, R. Cools, and D. Van Compernolle, "Template based continuous speech recognition," *IEEE Trans. on ASLP*, 2007, to appear.

[2] M. De Wachter, K. Demuynck, and D. Van Compernolle, "Outlier correction for local distance measures in example based speech recognition," in *Proc. ICASSP*, Honolulu, HI, U.S.A., Apr. 2007, accepted for publication.

[3] H. Ney, "Modeling and search in continuous speech recognition," in *Proc. EUROSPEECH*, vol. I, Berlin, Germany, September 1993, pp. 491–498.

[4] A. J. Hunt and A. W. Black, "Unit selection in a concatenative speech synthesis system using a large speech database," in *Proc. ICASSP*, vol. I, Atlanta, May 1996, pp. 373–376.

[5] K. Demuynck, J. Duchateau, and D. Van Compernolle, "Optimal feature sub-space selection based on discriminant analysis," in *Proc. EUROSPEECH*, vol. III, Budapest, Hungary, Sept. 1999, pp. 1311–1314.

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

[7] M. De Wachter, K. Demuynck, D. Van Compernolle, and P. Wambacq, "Data driven example based continuous speech recognition," in *Proc. EUROSPEECH*, Geneva, Switzerland, Sept. 2003, pp. 1133–1136.