



# A Segment-Based Non-Parametric Approach for Monophone Recognition

Ladan Golipour and Douglas O'Shaughnessy

INRS-EMT, Montreal, Canada

golipour@emt.inrs.ca, dougo@emt.inrs.ca

## Abstract

In this paper, we propose a segment-based non-parametric method of monophone recognition. We pre-segment the speech utterance into its underlying phonemes using a group-delay-based algorithm. Then, we apply the  $k$ -NN/SASH phoneme classification technique to classify the hypothesized phonemes. Since phoneme boundaries are already known during the decoding, the search space is very limited and the recognition fast. However, such hard-decisioning leads to missed boundaries and over-segmentations. Therefore, while constructing the graph for an utterance, we use phoneme duration constraints and broad-class similarity information to merge or split the segments and create new branches. We perform a simplified acoustical level monophone recognition task on the TIMIT test database. Since phoneme transitional probabilities are not included, only one (most likely) hypothesis and score is provided for each segment and a simple shortest path search algorithm is applied to find the best phoneme sequence rather than the Viterbi search. This simplified evaluation achieves 58.5% accuracy and 67.8% correctness.

**Index Terms:** phoneme recognition, nonparametric density estimation, phoneme segmentation

## 1. Introduction

Before the emergence of Hidden Markov Models (HMM)-based speech recognizers, dynamic time warping (DTW) was the basis of template-based speech recognition. However, the unacceptable computational time of this algorithm due to the large search space and its poor performance for speaker-independent tasks due to a lack of generalization lead to the decline of template-based approaches. Recently, there has been considerable progress in the design of fast and efficient search algorithms for pattern recognition problems. In addition, the growing access to cheap hardware helped researchers to reinvestigate nonparametric approaches. The speech recognition method proposed by Wachter et al in 2007 [12] was based on straight-forward template matching. They also take advantage of the discriminant nature of the  $k$ -NN classification rule. In 2003, Lefevre [7] integrated the  $k$ -NN classifier into an HMM-based recognition system, which led to the development of a baseline  $k$ NN/HMM system. Moreover, researchers incorporate discriminative techniques to the task of speech recognition. They apply prob-

ability density estimations with local discriminative ability, such as Neural Networks, or use discriminative training procedures, such as Maximum Mutual Information (MMI)-based methods [7]. However, the running-time of these techniques is large and they are expensive to implement.

In the proposed monophone recognition algorithm, instead of frame-based processing of speech, we detect the phoneme boundaries using our previously proposed phoneme segmentation algorithm and then decode these variable-length segments using the  $k$ -NN/SASH phoneme classifier. The main advantage of our recognition approach is its very small search space compared to other approaches, due to the limited number of phonemes in an utterance. In addition, since we apply a nonparametric density estimation, there is no training stage in the algorithm. These advantages make the recognition process very fast. During the decoding stage, we apply a procedure to compensate for missed boundaries and over-segmentations. For the evaluation of this algorithm only the acoustic scores are used. Therefore, the provided results should be compared to methods of similar states.

The outline of this paper is as follows. In Section 2, we provide an overview of the phoneme boundary detection algorithm. Next, we explain the  $k$ -NN classification technique and the similarity search algorithm, which we employ during the decoding stage in Section 3. In Section 4, we describe how feature vectors are generated for segments. Section 5 demonstrates the construction of a phonetic graph for an utterance and the missed boundary and over-segmentation compensation steps. In Section 6, we explain the score computation of the hypothesized phonemes and provide the results. Finally, we end the paper with some conclusions in Section 7. The evaluation of the algorithms is done using the clean TIMIT database. We use the first directory of the test database as the development set.

## 2. Phoneme Segmentation

In order to segment the utterance into the underlying phonemes, we apply a similar algorithm to our previously proposed phoneme segmentation algorithm, which is described in [3]. We compute 5 spectral change functions based on the gradient of the logarithm of the spectral energy in 5 separate frequency bands. The subbands are the first 4 formant bandwidths and one subband for high frequencies. We also incorporate a smoothing technique that is based on

the modified group-delay function before the peak-picking step. This function was defined and used by Murthy [8] and Nagarajan [9]. Due to the additive property of the group-delay function, the spectrum of modified group-delay function has a higher resolving capability compared to the magnitude spectrum. In addition, it can be applied on speech signals even if they show nonminimum phase property. We smooth out the overall fluctuations of the signal and at the same time preserve adjacent major peaks through applying the modified group-delay function. Processing independent frequency subbands has been a popular approach for other speech recognition architectures such as neural networks. In multi-net schemes, 5 frequency bands is the optimal number of subbands to be joined together for a broad-band speech such as TIMIT [10]. Finally, we use a threshold to select the peaks in the spectral change functions as the final phoneme boundaries. Applying the above phoneme boundary detection algorithm on the TIMIT test database resulted in 83.5% correct detection of phoneme boundaries and 18.6% over-segmentation.

### 3. Phoneme Classification

After segmenting the utterances into hypothesized phonemes, we model the phoneme sequence as a graph for which the nodes are assumed to be the positions of hypothesized phoneme boundaries. During the decoding stage, the nodes are linked together and phoneme labels and scores are assigned to the branches. In order to classify the segments confined between two nodes, we use an approximate  $k$ -NN classification scheme and the SASH similarity search algorithm.

#### ***k*-Approximate Nearest Neighbour Classification**

As mentioned before, researchers attempt to incorporate the *discriminant* density estimation in the recognition process. We employ a simple and intuitive *discriminant* probability estimate, the “volumetric”  $k$ -NN classification rule [1]. However, since we are employing a high dimensional feature representation for phoneme segments, in order to avoid the sparsity of the data space, we use the *approximate* nearest neighbours rather than exact ones. Fortunately, for real data, the accuracy of the approximate methods does not degrade considerably compared to the amount of savings in the computational time [5]. An approximate nearest neighbour of point  $q$  for the error bound  $\epsilon > 0$  is a point for which

$$\text{dist}(q, u) \leq (1 + \epsilon)r. \quad (1)$$

In this equation,  $u$  is the approximate nearest neighbour of  $q$ , and  $r$  is the true distance between the query point  $q$  and its  $k^{\text{th}}$  nearest neighbour.

#### **SASH Search Algorithm**

The time complexity of most of  $k$ -NN search algorithms is exponential in the number of dimensions. This fact has limited the use of this classifier and forced researchers to reduce the dimensionality through other computationally expensive methods, such as Linear Discriminant Analysis (LDA) or

MIDA [11]. However, in 2005, Houle proposed a similarity search technique called SASH for which the computational time are not affected by the dimensionality of the data, except for the distance computation. Also, through using the approximate similarity search, it performs the query search 2 orders of magnitude faster than other sequential search methods. We use this search algorithm in combination with the  $k$ -NN rule for classifying hypothesized segments.

In this algorithm, at first a SASH is constructed from the training datapoints. In order to construct the SASH, the datapoints are ordered randomly in a hierarchy in such a way that each level has twice as many points as the level above it. Next, the points of one level are connected to the points of the level above and below by determining their  $p$  *tentative parents* and their  $c$  *distinctive children*, respectively.

The *tentative parents* of a point are the closest nodes from the level above that are the *distinctive children* of the level above them. The *distinctive children* of a point are the closest nodes from level below that have chosen this point as their *tentative parent*. The proper choices for  $p$  and  $c$  are  $p = 4$  and  $c = 4p$ .

Once the SASH structure is completed, it can be used to search for the nearest neighbours of any query point. For this purpose, certain numbers of points are selected from each level of the SASH. These points are the closest neighbours of the query point among the distinctive children of the level above. Finally,  $k$  closest points to the query point are drawn from the union of the selected points from all levels. These are the final nearest neighbours of the unknown pattern. The time complexity of constructing the SASH structure is  $(2p^2n \log_2 n)$  and for the  $k$ -ANN query search is  $\left(\frac{k^{1+\frac{1}{\log_2 n}}}{k^{\frac{1}{\log_2 n}-1}} + 2p^3 \log_2 n\right)$ .  $n$  is the total number of training points

### 4. Phoneme Feature Extraction

In order to avoid the computational time of the time-warping distance computation, we use a concatenated fixed-length feature vector [6]. However, we reduce the number of averaged frames and use a higher dimensional feature space to induce more discrimination between phoneme classes. We divide each of the 3 sections in 2 and concatenate the 42 dimensional MFCCs of 6 sections. The total number of features including the duration of the phoneme reaches 253. We examined larger dimensionalities. However since the amount of training data does not increase relatively, too sparse a feature space can occur, which decreases the classification rate.

### 5. Graph Construction

If the phoneme segmentation could be performed with complete accuracy, the problem of phoneme recognition would be reduced to phoneme classification. However, missed boundaries and over-segmentations occur in all methods of phoneme or syllable segmentation and only a trade-off be-

tween the two is possible. Therefore, during the construction of the graph, we take advantage of the extra peaks and use the duration constraints to split long segments or the broad-class similarity information to combine the short segments. Then, we add new branches to the graph for the new segments produced during the compensation stage. After the graph is completed, we apply a shortest path graph search method, such as Dijkstra’s algorithm, to find the phoneme sequence with the minimum cost.

We assign one node to every selected boundary and call these nodes set  $A$ . We also assign one node to every other non-zero boundary marked by the phoneme segmentation algorithm before the final selection and call them set  $B$ . During the operation described below, by *connecting* two nodes or *creating* a branch between them, we mean (1) classifying the segment between the two boundaries associated with those nodes, (2) computing the score for the phoneme hypothesis, and (3) assigning this information to the branch between the two nodes.

**Step 1:** For every two adjacent members of set  $A$ , starting from the first member, we repeat the following steps:

- We *connect* the two nodes in the graph.
- We call the timing position of the first node  $bndL$  and the timing position of the second node  $bndR$ .
- We define a lower limit, **LimDur.L**, for the duration of the phoneme using mean ( $\mu_{Dur}$ ) and standard deviation ( $\sigma_{Dur}$ ) of the duration of the phoneme:  

$$\text{LimDur.L}(\text{phoneme}) = \mu_{Dur}(\text{phoneme}) - \sigma_{Dur}(\text{phoneme})$$
The mean and standard deviation of the duration of all phoneme classes were calculated during the training process.
- We use members of set  $B$  that fall between  $bndL$  and  $bndR$  (e.g.,  $bndx$ ,  $bndy$ ) to split the phoneme as below.

**Missed Boundary Compensation**

- If  $(bndR - bndL) \geq 1 \times \text{LimDur.L}(\text{phoneme})$ , we *create* paths with structure  $bndL$ - $bndx$ - $bndR$ .
- If  $(bndR - bndL) \geq 2 \times \text{LimDur.L}(\text{phoneme})$ , we *create* paths with structure  $bndL$ - $bndx$ - $bndy$ - $bndR$ .
- If  $(bndR - bndL) \geq 3 \times \text{LimDur.L}(\text{phoneme})$ , we *create* paths with structure  $bndL$ - $bndx$ - $bndy$ - $bndz$ - $bndR$ .

...

**Step 2:** Once the graph is constructed in step 1, for every two branches that share one node, e.g.  $bndm$ - $bndl$  and  $bndl$ - $bndn$ , we conduct the following procedure:

**Over-Segmentation Compensation**

- If the hypothesized phonemes of the two branches have a similar broad class (vowel, fricative, ...), there is a possibility that they are parts of a single phoneme. We define an upper limit for the duration of the phoneme and examine the duration of the two branches (phoneme1 and phoneme2) as below:  

$$\text{LimDur.U}(\text{phoneme}) = \mu_{Dur}(\text{phoneme}) + \sigma_{Dur}(\text{phoneme})$$
- If  $(bndn - bndm) \leq (\text{LimDur.U}(\text{phoneme1}) + \text{LimDur.U}(\text{phoneme2}))$ , we combine the two branches,

Table 1: The recognition results for the missed boundary compensation and over-segmentation steps on the development set.

Method	Correctness	Accuracy
No missed boundary comp.	60.7%	55.3%
No over-segmentation comp.	69.6%	53.1%
With boundary comp.	68.1%	58.1%

meaning that we *connect* the exterior nodes,  $bndm$  and  $bndn$ , together.

Figure 1 demonstrates an example of how missed boundaries and over-segmentations are compensated for. Also, in order to display the effect of the compensating steps, we perform the recognition once without the missed boundary compensation step and once without the over segmentation compensation step. The recognition results of these cases on the development set are depicted in Table 1. Basically, in order to compensate for missed boundaries and over-segmentations, we separate the two problems. We first assume that all boundaries in set  $A$  are true phoneme boundaries and there is only the problem of missed boundaries. Therefore, we search between each two adjacent peaks in set  $A$  to find new phoneme boundaries that satisfy the duration constraint. Next, we assume that the constructed graph includes true boundaries and extra ones. Therefore, we continue by combining those segments that can potentially belong to one phoneme.

## 6. Phoneme Score Computation

The related score for the recognized phoneme is computed using the  $k$ -NN density estimation [1]:

$$\text{Score}(bndL, bndR) = (bndR - bndL) \times \log\left(\frac{k}{R_k^d \times n}\right).$$

$R_k^d$  is the distance between the query point and its  $k^{th}$  nearest neighbour,  $d$  is the dimensionality of the feature space, and  $n$  is the total number of training points. We multiply the  $k$ -NN score by the duration of the phoneme in order to be able to compare the total score of different paths with each other, otherwise the number of phonemes in a path would affect the score of that path. The effect of the dimensionality on the recognition performance is displayed in Table 2. Since the recognition algorithm is a trial version, we examine its acoustical level performance. This means that phoneme transitional probabilities or contextual information are not employed during graph construction. Therefore, the best phoneme sequence would be the one for which the summation of acoustic scores is maximum. As a result, one phoneme hypothesis and score is used for each segment, and the Viterbi search is substituted with a simple shortest path search algorithm. We intend to enhance our algorithm by including the higher-level linguistic information and applying the Viterbi search technique. The phoneme recognition performance of our system for TIMIT

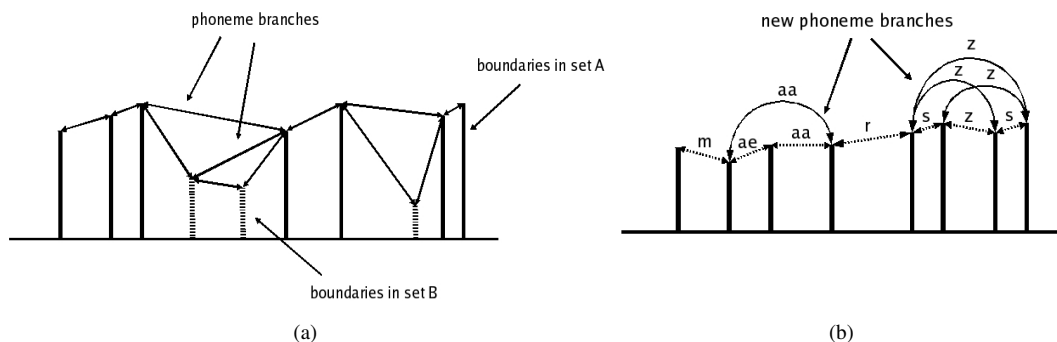


Figure 1: (a) An example for the missed boundaries compensation. (b) An example for the over-segmentation compensation.

test database is correctness of 67.8% and accuracy of 58.5%. Unfortunately, not many methods provide results of a similar state since the higher level linguistic information provides a considerable improvement on the recognition result. The accuracy of a traditional context-dependent GMM-HMM monophone recognizer drops from 64.4% to 53.7% if only the context-independent case is used [4]. The baseline

missed boundaries and over-segmentations through incorporating phoneme broad-class and duration constraints. The proposed algorithm could achieve the accuracy of 58.5% along with 67.8% correctness for phoneme recognition on the TIMIT test database, using only the acoustic information of speech.

Table 2: The performance of the proposed monophone recognition for different dimensions on the development set.

	$1 \times 42$	$3 \times 42$	$6 \times 42$	$9 \times 42$
Correctness(%)	51.9	57.4	<b>68.1</b>	70.6
Accuracy(%)	49.1	54.1	<b>58.1</b>	52.6

$k$ NN/HMM system by Lefevre [7] obtained an accuracy of 52.5% with 57.6% correctness for the context-independent task on the TIMIT test database. The phoneme recognition rate of Wachter's [11] template-based system for a full context-dependent model was 70.4%. Finally, Glass [2] proposed a segmental-based phoneme recognizer and achieved an accuracy of 64.1% for the context-independent case. All these systems at least use phoneme transitional probabilities. As mentioned before, the proposed segment-based  $k$ -NN/SASH recognition algorithm does not require to train model parameters. Also, the use of phoneme boundary constraints leads to a small search space compared to frame-based techniques, and in an enhanced version, when the Viterbi search is used, the computational time can be saved and the error propagation can be limited. In addition, the computational time of the SASH search algorithm does not depend on the dimensionality of the data; making it possible to use high-dimensional feature vectors and avoid the time-consuming DTW-based distance computation.

## 7. Conclusion

In this paper, we propose a segment-based  $k$ -NN/SASH monophone recognition algorithm that is based on the pre-segmentation of utterances into their underlying phonemes. Both the segmentation and classification algorithms in this approach can be applied independently and compete favorably with state-of-the-art methods. While generating the graph for the utterance, we attempt to compensate for the

## 8. References

- [1] Fukunaga K., "Introduction to statistical pattern recognition", *Morgan Kaufmann*, 1990, edition 2.
- [2] Glass J. R., "A probabilistic framework for segment-based speech recognition", *Journal of Computer Speech and Language*, 2003, vol. 17, no. 2, pp. 137-152.
- [3] Golipour L., and O'Shaughnessy D., "A new approach for phoneme segmentation of speech signals", *Proc. Eurospeech*, 2007, pp. 1933-1936.
- [4] Graves A., Fernández S., and Schmidhuber Jürgen, "Bidirectional LSTM networks for improved phoneme classification and recognition", *Proc. ICANN*, 2005, pp. 602-610.
- [5] Houle M. E., and Sakuma J., "Fast approximate similarity search in extremely high-dimensional datasets", *Proc. 21st International Conference on Data Engineering (ICDE)*, 2005, pp. 619-630.
- [6] Lee S., and Glass J., "Real-time probabilistic segmentation for segment-based speech recognition", *Proc. ICSLP*, 1998, pp. 1641-1648.
- [7] Lefevre F., "Non-parametric probability estimation for HMM-based automatic speech recognition", *Computer, Speech and Language*, 2003, vol. 17, no. 2-3, pp. 113-136.
- [8] Murthy H. A., and Gadde V., "The modified group delay function and its application to phoneme recognition," *Proc. ICASSP*, 2003, vol. 1, pp. 68-71.
- [9] Nagarajan T., and Murthy H. A., "Subband-based group delay segmentation of spontaneous speech into syllable-like units," *EURASIP Journal on Applied Signal Processing*, 2004, vol. 17, pp. 2614-2625.
- [10] Schwarz P., Matejka P., and Cernocky J., "Hierarchical structures of neural networks for phoneme recognition", *Proc. ICASSP*, 2006, pp. 325-328.
- [11] De Wachter M., "Example based continuous speech recognition", *PhD thesis, K.U.Leuven, ESAT*, May 2007.
- [12] De Wachter M., Matton M., Demuyneck K., Wambacq P., Cools R., and van Compernelle D., "Template-based continuous speech recognition", *IEEE Transactions on Audio, Speech, and Language Processing*, 2007, vol. 15, no. 4, pp. 1377-1390.