
Phoneme Alignment using Large Margin Techniques

Joseph Keshet Shai Shalev-Shwartz Yoram Singer
School of Computer Science & Engineering
The Hebrew University, Jerusalem 91904, Israel
{jkeshet,shais,singer}@cs.huji.ac.il

Phoneme alignment is the task of proper positioning of a sequence of phonemes in relation to a corresponding continuous speech signal. This problem is also referred to as phoneme segmentation. An accurate and fast alignment procedure is a necessary tool for developing speech recognition and text-to-speech systems.

We propose an alignment method which is based on recent advances in kernel machines and large margin classifiers for sequences [13, 12], which in turn build on the pioneering work of Vapnik and colleagues [15, 4]. The alignment function we devise is based on mapping the speech signal and its phoneme representation along with the target alignment into an abstract vector-space. Building on techniques used for learning SVMs, our alignment function distills to a classifier in this vector-space which is aimed at separating correct alignments from incorrect ones. We describe a simple iterative algorithm for learning the alignment function and discuss its formal properties. Experiments with the TIMIT corpus show that our method outperforms the best performing HMM-based approach [1].

In the alignment problem, we are given a speech utterance along with a phonetic representation of the utterance. Our goal is to generate an alignment between the speech signal and the phonetic representation. The Mel-frequency cepstrum coefficients (MFCC) along with their first and second derivatives are extracted from the speech in the standard way which is based on the ETSI standard for distributed speech recognition. We denote the domain of the acoustic feature vectors by $\mathcal{X} \subset \mathbb{R}^d$. The acoustic feature representation of a speech signal is therefore a sequence of vectors $\bar{\mathbf{x}} = (\mathbf{x}_1, \dots, \mathbf{x}_T)$, where $\mathbf{x}_t \in \mathcal{X}$ for all $1 \leq t \leq T$. A phonetic representation of an utterance is defined as a string of phoneme symbols. Formally, we denote each phoneme by $p \in \mathcal{P}$, where \mathcal{P} is the set of 48 English American phoneme symbols as proposed by [8]. Therefore, a phonetic representation of a speech utterance consists of a sequence of phoneme values $\bar{p} = (p_1, \dots, p_k)$. Note that the number of phonemes clearly varies from one utterance to another and thus k is not fixed. We denote by \mathcal{P}^* (and similarly \mathcal{X}^*) the set of all finite-length sequences over \mathcal{P} . In summary, an alignment input is a pair $(\bar{\mathbf{x}}, \bar{p})$ where $\bar{\mathbf{x}}$ is an acoustic representation of the speech signal and \bar{p} is a phonetic representation of the same signal. An alignment between the acoustic and phonetic representations of a spoken utterance is a sequence of start-times $\bar{y} = (y_1, \dots, y_k)$ where $y_i \in \mathbb{N}$ is the start-time (measured as frame number) of phoneme i in the acoustic signal. Each phoneme i therefore starts at frame y_i and ends at frame $y_{i+1} - 1$.

Clearly, there are different ways to pronounce the same utterance. Different speakers

have different accents and tend to speak at different rates. Our goal is to learn an alignment function that predicts the true start-times of the phonemes from the speech signal and the phonetic representation.

Most previous work utilized hidden Markov models for solving the alignment problem. In this paper we describe and analyze an alternative paradigm in which the learning phase is tightly coupled with the decision task the algorithm must perform. Rather than working with probability functions we assume the existence of a predefined set of base alignment functions, $\{\phi_j\}_{j=1}^n$. Each base function takes the form $\phi_j : \mathcal{X}^* \times (\mathcal{P} \times \mathbb{N})^* \rightarrow \mathbb{R}$. Thus, the input of each base function is an acoustic-phonetic representation, $(\bar{\mathbf{x}}, \bar{p})$, together with a candidate alignment \bar{y} . The base function returns a scalar which, intuitively, represents the confidence in the suggested alignment, \bar{y} . We denote by $\phi(\bar{\mathbf{x}}, \bar{p}, \bar{y})$ the vector in \mathbb{R}^n whose j th element is $\phi_j(\bar{\mathbf{x}}, \bar{p}, \bar{y})$. The alignment functions we use are of the form

$$f(\bar{\mathbf{x}}, \bar{p}) = \operatorname{argmax}_{\bar{y}} \mathbf{w} \cdot \phi(\bar{\mathbf{x}}, \bar{p}, \bar{y}) \quad , \quad (1)$$

where $\mathbf{w} \in \mathbb{R}^n$ is a vector of importance weights that must be learned. In words, f returns a suggestion for an alignment sequence by maximizing a weighted sum of the scores returned by each base function ϕ_j . Note that the number of possible alignment sequences is exponentially large. Nevertheless, as in the generative case, if the base functions ϕ_j are decomposable, the optimization in Eq. (1) can be efficiently calculated using a dynamic programming procedure.

As mentioned above, we would like to learn the function f from examples. Each example is composed of an acoustic and a phonetic representation of an utterance $(\bar{\mathbf{x}}, \bar{p})$ together with the true alignment between them, \bar{y} . Let $\bar{y}' = f(\bar{\mathbf{x}}, \bar{p})$ be the alignment suggested by f . We denote by $\gamma(\bar{y}, \bar{y}')$ the cost of predicting the alignment \bar{y}' where the true alignment is \bar{y} . Formally, $\gamma : (\mathbb{N} \times \mathbb{N})^* \rightarrow \mathbb{R}$ is a function that gets two alignments and returns a scalar which is the cost of predicting \bar{y}' where the true alignment is \bar{y} . We assume that $\gamma(\bar{y}, \bar{y}') \geq 0$ and that $\gamma(\bar{y}, \bar{y}) = 0$. An example for such a cost function is, $\gamma(\bar{y}, \bar{y}') = \frac{1}{|\bar{y}|} \sum_{i=1}^{|\bar{y}'|} |y_i - y'_i|$. The above cost is the average of the absolute differences between the predicted alignment and the true alignment. In our experiments, we used a variant of the above cost function and replaced the summands $|y_i - y'_i|$ with $\max\{0, |y_i - y'_i| - \varepsilon\}$, where ε is a predefined small constant. The advantage of this cost is that no loss is incurred due to the i th phoneme if y_i and y'_i are within a distance ε of each other. The goal of the learning process is to find an alignment function f that attains small cost on unseen examples.

We now show how to use the training set in order to find an alignment function f which with high probability, attains a small cost on the training set and on unseen examples as well. Recall that our construction is based on a set of base alignment functions $\{\phi_j\}_{j=1}^n$ which maps an acoustic-phonetic representation of a speech utterance as well as a suggested alignment into an abstract vector-space. We utilize seven different base functions ($n = 7$). These base functions are used for defining our alignment function $f(\bar{\mathbf{x}}, \bar{p})$ as in Eq. (1). To facilitate an efficient evaluation of $f(\bar{\mathbf{x}}, \bar{p})$ we enforce structural constraints on the base functions.

A supervised learning algorithm for alignment receives as input a training set $S = \{(\bar{\mathbf{x}}_1, \bar{p}_1, \bar{y}_1), \dots, (\bar{\mathbf{x}}_m, \bar{p}_m, \bar{y}_m)\}$ and returns a weight vector \mathbf{w} defining the alignment function f given by Eq. (1). Similar to the SVM algorithm for binary classification, our approach for choosing the weight vector \mathbf{w} is based on the idea of large-margin separation. However, in our case, alignments are not merely correct or incorrect. Instead, the cost function $\gamma(\bar{y}, \bar{y}')$ is used for assessing the quality of alignments. Therefore, we do not aim at separating correct alignments from incorrect ones but rather try to rank alignments according to their quality. Theoretically, our approach

can be described as a two-step procedure: First, we construct a vector $\phi(\bar{\mathbf{x}}_i, \bar{p}_i, \bar{y}')$ in the vector space \mathbb{R}^n based on each instance $(\bar{\mathbf{x}}_i, \bar{p}_i)$ in the training set S and each possible alignment \bar{y}' . Second, we find a vector $\mathbf{w} \in \mathbb{R}^n$, such that the projection of vectors onto \mathbf{w} ranks the vectors constructed in the first step above according to their quality. Formally, for each instance $(\bar{\mathbf{x}}_i, \bar{p}_i)$ and for each possible suggested alignment \bar{y}' , the following constraint should hold,

$$\mathbf{w} \cdot \phi(\bar{\mathbf{x}}_i, \bar{p}_i, \bar{y}_i) - \mathbf{w} \cdot \phi(\bar{\mathbf{x}}_i, \bar{p}_i, \bar{y}') \geq \gamma(\bar{y}_i, \bar{y}') - \xi_i, \quad (2)$$

where ξ_i is a non-negative slack variable indicates the loss of the i th example. The SVM solution for the problem is therefore the weight vector $\mathbf{w} \in \mathbb{R}^n$ which minimizes the objective function $\frac{1}{2}\|\mathbf{w}\|^2 + C \sum_{i=1}^m \xi_i$ while satisfying all the constraints in Eq. (2). The parameter C serves as a complexity-accuracy trade-off parameter (see [4]).

In practice, the above two-step procedure can not be directly implemented since the number of constraints is exponentially large. To overcome this obstacle, we describe a simple iterative procedure for finding \mathbf{w} . Our iterative algorithm first constructs a sequence of weight vectors $\mathbf{w}_0, \mathbf{w}_1, \dots, \mathbf{w}_m$. The first weight vector is set to be the zero vector, $\mathbf{w}_0 = \mathbf{0}$. On iteration i of the algorithm, we utilize the i th example of the training set along with the previous weight vector \mathbf{w}_i , for defining the next weight vector \mathbf{w}_{i+1} . Let \bar{y}' be the predicted alignment sequence for the i th example according to \mathbf{w}_i . We set the next weight vector \mathbf{w}_{i+1} to be the minimizer of the following optimization problem,

$$\begin{aligned} \min_{\mathbf{w} \in \mathbb{R}^n, \xi \geq 0} \quad & \frac{1}{2}\|\mathbf{w} - \mathbf{w}_i\|^2 + C\xi \quad \text{s.t.} \\ & \mathbf{w} \cdot \phi(\bar{\mathbf{x}}, \bar{p}, \bar{y}) - \mathbf{w} \cdot \phi(\bar{\mathbf{x}}, \bar{p}, \bar{y}') \geq \sqrt{\gamma(\bar{y}, \bar{y}') - \xi} \quad . \end{aligned} \quad (3)$$

This optimization problem can be thought of as a relaxed version of the SVM optimization problem with three major differences. First, we replace the exponential number of constraints from Eq. (2) with a single constraint. This constraint is based on the predicted alignment \bar{y}' according to the previous weight vector \mathbf{w}_i . Second, we replaced the term $\|\mathbf{w}\|^2$ in the objective function of the SVM with the term $\|\mathbf{w} - \mathbf{w}_i\|^2$. Intuitively, we would like to minimize the loss of \mathbf{w} on the current example, i.e., the slack variable ξ , while remaining as close as possible to our previous weight vector \mathbf{w}_i . Last, we replace $\gamma(\bar{y}, \bar{y}')$ with $\sqrt{\gamma(\bar{y}, \bar{y}')}$ for technical reasons which will be given elsewhere. It can be shown (see [3]) that the solution to the above optimization problem is, $\mathbf{w}_{i+1} = \mathbf{w}_i + \min\{\ell/\|\mathbf{a}\|^2, C\} \mathbf{a}$, where $\mathbf{a} = \phi(\bar{\mathbf{x}}_i, \bar{p}_i, \bar{y}_i) - \phi(\bar{\mathbf{x}}_i, \bar{p}_i, \bar{y}')$ and $\ell = \max\{(\gamma(\bar{y}_i, \bar{y}'))^{1/2} - \mathbf{w}_i \cdot \mathbf{a}, 0\}$.

The above iterative procedure gives us a sequence of weight vectors. We briefly note that it has been proved that at least one of the resulting alignment functions is likely to have good generalization properties [12, 2]. To find an alignment function that generalizes well, we calculate the average cost of each alignment function on a validation set and choose the one that achieves the best results.

To validate the effectiveness of the proposed approach we performed experiments with the TIMIT corpus. We used the training portion of TIMIT for learning an alignment function. We evaluated the learned alignment function on both the core test set and the entire test set of TIMIT. A comparison of our results with the results reported in [1] is provided in Tab. . For each tolerance value $\tau \in \{10 \text{ ms}, 20 \text{ ms}, 30 \text{ ms}, 40 \text{ ms}\}$, we counted the number of predictions whose distance to the true boundary, $t = |y_i - y'_i|$, is less than τ . As can be seen, our discriminative method outperforms the generative approach described in [1] on all predefined tolerance values.

	Test size	$t \leq 10$ ms	$t \leq 20$ ms	$t \leq 30$ ms	$t \leq 40$ ms
Discrim. Alignment	192	79.7	92.1	96.2	98.1
Brugnara <i>et al</i> [1]	192	75.3	88.9	94.4	97.1
Discrim. Alignment	1344	80.0	92.3	96.4	98.2
Brugnara <i>et al</i> [1]	1344	74.6	88.8	94.1	96.8

Table 1: Percentage of correctly positioned boundaries, given a predefined tolerance

References

- [1] F. Brugnara, D. Falavigna, and M. Omologo. Automatic segmentation and labeling of speech based on hidden Markov models. *Speech Comm.*, 12:357–370, 1993.
- [2] N. Cesa-Bianchi, A. Conconi, and C. Gentile. On the generalization ability of on-line learning algorithms. In *NIPS*, 2002.
- [3] K. Crammer, O. Dekel, S. Shalev-Shwartz, and Y. Singer. Online passive aggressive algorithms. In *NIPS*, 2003.
- [4] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines*. Cambridge Univ. Press, 2000.
- [5] O. Dekel, J. Keshet, and Y. Singer. Online algorithm for hierarchical phoneme classification. In *MLMI*, 2004.
- [6] J.-P. Hosom. Automatic phoneme alignment based on acoustic-phonetic modeling. In *ICSLP*, 2002.
- [7] J. Keshet, D. Chazan, and B.-Z. Bobrovsky. Plosive spotting with margin classifiers. In *EUROSPEECH*, 2001.
- [8] K.-F. Lee and H.-W. Hon. Speaker independent phone recognition using hidden Markov models. *IEEE Trans. Acous., Speech. and Signal Processing*, 37(2):1641–1648, 1989.
- [9] Z. Litichever and D. Chazan. Classification of transition sounds with application to automatic speech recognition. In *EUROSPEECH*, 2001.
- [10] L. Rabiner and B.H. Juang. *Fundamentals of Speech Recognition*. Prentice Hall, 1993.
- [11] J. Salomon, S. King, and M. Osborne. Framewise phone classification using support vector machine. *ICSLP*, 2002.
- [12] S. Shalev-Shwartz, J. Keshet, and Y. Singer. Learning to align polyphonic music. In *ISMIR*, 2004.
- [13] B. Taskar, C. Guestrin, and D. Koller. Max-margin Markov networks. In *NIPS 17*, 2003.
- [14] D.T. Toledano, L.A.H. Gomez, and L.V. Grande. Automatic phoneme segmentation. *IEEE Trans. Speech and Audio Proc.*, 11(6):617–625, 2003.
- [15] V. N. Vapnik. *Statistical Learning Theory*. Wiley, 1998.