Event Selection from Phone Posteriorgrams Using Matched Filters

Keith Kintzley[1], *Aren Jansen*[1,2], *Hynek Hermansky*[1,2]

[1]Dept. of Electrical and Computer Engineering, [2]HLT Center of Excellence
Johns Hopkins University, Baltimore, Maryland
kintzley@jhu.edu, aren@jhu.edu, hynek@jhu.edu

Abstract

In this paper we address the issue of how to select a minimal set of phonetic events from a phone posteriorgram while minimizing the loss of information. We derive phone posteriorgrams from two sources, Gaussian mixture models and sparse multilayer perceptrons, and apply phone-specific matched filters to the posteriorgrams to yield a smaller set of phonetic events. We introduce a mutual information based performance measure to compare phonetic event selection techniques and demonstrate that events extracted using matched filters can reduce input data while significantly improving performance of an event-based keyword spotting system.

Index Terms: matched filters, keyword spotting, point process model, posteriorgram

1. Introduction

In most recognition systems, speech signal acoustic features are used to infer a sequence of hidden phonetic states for each frame of the signal. One representation of such information is provided by the phone posteriorgram which expresses the posterior probability of every phone given the acoustic features for each frame of speech. Given that the message produced by the speaker is a discrete sequence of phones, then it is clear the posteriorgram representation contains surplus information. The specific recognition task that we have in mind is keyword spotting, which is concerned with searching through large volumes of acoustic speech data to locate all instances of a specific term contained in a collection. In contrast to a full speech recognition system, we are not attempting to reconstruct the entire sequence of words and operate without the assistance of a language model. Indeed, if only a few terms are of interest in a large volume of speech, then complete transcription is unnecessary and likely to be computationally infeasible. In such a setting, the speed of the search is paramount and for the event-based keyword spotting considered here, fast processing can be obtained by minimizing the set of phonetic events.

Historically, keyword spotting systems have been constructed using hidden Markov models composed of keyword, background and filler submodels operating on frame-by-frame acoustic vectors. In such systems, keyword detections are declared when the Viterbi best-path state sequence passes through the keyword model (see [1]). An alternative approach is the event-based model, an early example of which is found in [2]. This statistical model uses auditory events which are focused on perceptually dominant portions of the signal, and these information-rich events were shown to be more robust to adverse acoustic conditions. Also, since the auditory events occurred every 50-150 ms [2], as opposed to every 10 ms frame, they provided a large data reduction. In a novel keyword spotting system presented in [3], data reduction is accomplished through phonetic matched filters which reduce a phone posteriorgram to a sequence of phonetic events for the task of detecting digits. More recently in [4], an event-based keyword spotting system based on Poisson process models was introduced and shown to be competitive with HMM systems, and this will serve as our evaluation system.

In this work, we combine the approaches of [3] and [4] using matched filters to derive phonetic events for an event-based keyword spotting system. We begin with a discussion of phone posteriorgrams and strategies for identifying phonetic events using local maxima of the posterior trajectories. We propose a new strategy for obtaining a reduced set of phonetic events by applying phone-specific matched filters. In order to quantify the performance of these approaches, we introduce a measure based on mutual information. Finally, we discuss the Poisson process model for event-based keyword spotting and provide experimental results showing that performance improves with a minimal set of events.

2. Phonetic events

2.1. Phone posteriorgrams

For a given acoustic feature vector x, consider that we have a set of phone detectors $g_p(x)$ for each phone p in the set of all phones $\mathcal{P}$ which estimate the posterior probability of p given x. A posteriorgram refers to the vector time series of posterior distributions across the phone set as a function of time, and we refer to the posterior probability of a single phone as a function of time as a *posterior trajectory*. Here we will consider two systems for estimating phone posterior probabilities, one based on Gaussian mixtures models (GMM) and the other derived from multilayer perceptrons.

The GMM-based system detailed in [4] uses standard 39-dimensional Mel-frequency cepstral coefficients for each speech frame with 8 mixture components (full covariance) to estimate $P(x|p)$ for each frame $x \in \mathbb{R}^{39}$ and $p \in \mathcal{P}$, and then computes posterior probabilities $P(p|x)$ using Bayes' rule. As an alternative to GMM-based posterior data, we introduce events based upon posteriorgrams produced by a sparse multilayer perceptron (SMLP) based system from [5]. In this system, perceptual linear prediction features are presented to an MLP which includes a sparse hidden layer and outputs 3-state phone posterior probabilities. Sparsity is enforced during training through the use of a sparse regularization term. A second MLP then converts the 3-state phone posteriors into single state phone posteriors. Both GMM and SMLP-based systems produce posteriorgram outputs, so phonetic events can be generated in the same manner.

2.2. Determining phonetic events

Given a posteriorgram representation, we now address the question of how to distill dense frame-by-frame posterior data into a sparse set of phonetic events. In the original presentation of point process models for keyword spotting [4], phonetic events were defined as the local maxima times of the raw posterior trajectories exceeding a threshold of $\delta = 0.5$. Here, the intuition was that probability one-half corresponds to the Bayes optimal binary classification decision. A simple example of the events derived in this manner is illustrated in Figure 1b. We designate these *local maxima* based events herein. For events defined in this manner, it is common for several local maxima to occur in the duration of a particular phone instance, but the sparsest representation would consist of just one event per phone. It should be pointed out that although these points appear to convey magnitude and timing information, for the model we will be considering *only the time of the event* is retained. Thus, the phonetic events for Figure 1b are $\{14, 16, 34, 37\}$, the frames at which the points occur.

To begin the analysis of alternative methods of determining phonetic events, let us consider for the moment the existence of ideal phone detectors whose outputs are either 0 or 1 and operate with 100% accuracy, perfectly matching phonetic labels for every frame. Given such detectors, for each phone trajectory we could define a phonetic event as the midpoint of the phone labels (see Figure 1a), which yields precisely one event per label. Any fewer points would imply a loss of phonetic information, so this set represents a lower bound on the number of events that we could hope to obtain. While such detectors don't exist, we can derive the set of events they would produce using phone label data. We will refer to this set as *oracle* events. While this represents the sparsest set of events, it is not immediately apparent that the point process keyword spotting system discussed below will perform well with such a limited set of points.

The posterior trajectories obtained from real detectors as shown in Figure 1b differ significantly from the ideal binary-valued trace shown in 1a. However, it is apparent that that both trajectories evidence the same underlying phonetic events. Considering the problem from the perspective of a communications system, the speaker of the utterance is transmitting information as a sequence of distinct phones which is converted into an acoustic signal. In a sense, our phone detector is a receiver, outputting (imperfectly) the presence of a particular phone at each frame, but what we really desire is the original phone string. In a manner similar to [3], if we consider the phone labels to be the clean signal and posterior output as noise corrupted, one mechanism for detecting the original symbols would be to apply matched filters. Since phone instances vary in duration, we obviously do not have a fixed signal for which to design a matched filter, so we consider the average signal profile instead. In [3], filters specific to each phone were obtained by averaging 0.5 sec windows of the *actual* posterior trajectory (Figure 1b) of all examples of the phone aligned to phone centers. In this work, we derived equivalent filters by instead averaging 0.5 sec windows of the *ideal* trajectory derived directly from directly from phone labels (Figure 1a). Figure 2 shows some filters resulting for a selection of phones. Given these filters, we then convolve each raw posterior trajectory with its corresponding filter to obtain a smoothed posterior trajectory as shown in Figure 1c. We then define *filtered* events as the local maxima of the smoothed trajectory exceeding a threshold δ, visually very similar to the oracle events in Figure 1a.

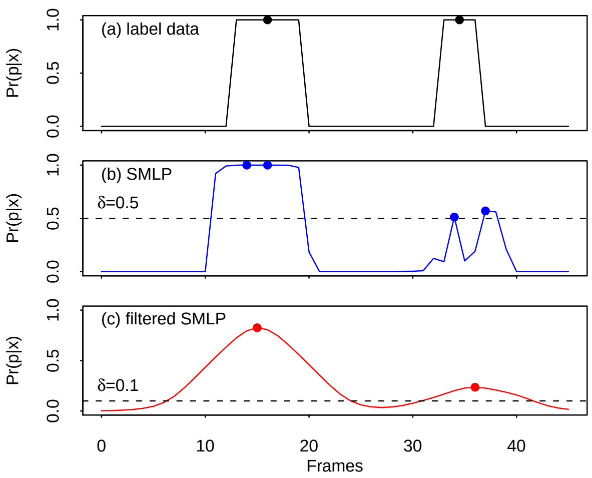

Figure 1: *Examples of posterior trajectories of phone /iy/. Symbols indicate the set of points which would be marked as events given the threshold δ and correspond to (a) oracle, (b) local maxima and (c) filtered events.*

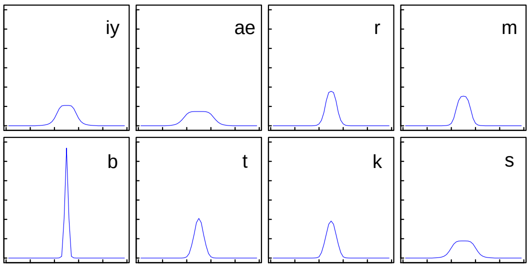

Figure 2: *Examples of matched filters for a selection of phones.*

2.3. Evaluating phonetic event selection techniques

Here we propose a metric to compare phone event selection techniques based upon mutual information between phone labels and the resulting phonetic events. We can think of our phone detectors and event selection mechanism as a noisy communications channel. Each channel input is a single phone (spanning successive frames) uttered by the speaker as indicated by the phone labeling. The channel output consists of all the phonetic events which occur during the span of the input phone. For the simplest case consider the oracle events for which there exists a single phonetic event for each input phone. For this ideal channel there is no loss of information, so the mutual information between the input and output distribution is just the entropy of the input.

For the non-ideal channel such as a posteriorgram with local maxima or filtered events, we must consider other output possibilities. First, it is possible that an input produces no output, so we will augment our set of outputs with an *erasure* (deletion) event. Next, a single input can produce multiple outputs (insertions) and we propose handling this with fractional counts. Consider a count matrix in which the rows are input phones and the columns are output phones and the erasure symbol. Suppose the phone /s/ is uttered resulting in phonetic events /s/,/s/, and /z/. In row /s/ of the count matrix we would record a count of 2/3 in the column corresponding to output /s/ and 1/3 in column /z/. This matrix corresponds to the joint distribution of input and output events, and from it we can compute mutual information.

So far we have not addressed the issue of setting the event threshold δ illustrated in Figure 1. As would be expected, using a high threshold produces many erasures and low mutual information. Alternatively, a very low threshold produces numerous false alarms also resulting in low mutual information. Thus, it is hoped that this measure will allow us to find good thresholds between these extremes.

2.4. Point Process Model

The keyword spotting system used in this work is based upon a point process model for keyword spotting described completely in [4]. In this framework, the input speech signal is represented by an extremely sparse set of phonetic events. Models built upon this representation take advantage of not only the identities of the phones detected, but also the sequence and relative timing between the events.

Given a keyword w and a set of observed phonetic events $O(t)$ beginning at time t, the output of the model is the detection function $d_w(t)$ given by

$$d_w(t) = \log\left[\frac{P(O(t)|\theta_w)}{P(O(t)|\theta_{bg})}\right], \quad (1)$$

where θ_w corresponds to the keyword-specific model parameters and θ_{bg} corresponds to background model parameters. This detection function is simply a log likelihood ratio evaluated at t which takes large values when it is likely that the keyword occurred. For each phone p in the set of all phones $\mathcal{P}$, we define $N_p = \{t_1, \cdots, t_{n_p}\}$, the set of points in time at which phone p occurs relative to time t. The observation $O(t) = \{N_p\}_{p\in\mathcal{P}}$ is thus the collection of these sets of points. Assuming for the moment a fixed keyword duration T, we will now specify the form of the models which yield estimates of $P(O(t)|T, \theta_w)$ and $P(O(t)|T, \theta_{bg})$.

For both cases, we model the set of points as having arisen from underlying Poisson processes, the background model being homogeneous and the keyword model as inhomogeneous. First we consider the homogeneous case of the background model. Under the simplifying assumption that the Poisson process for each phone p is independent of other phones, we can then express the likelihood of the entire collection of points $O(t)$ under the background model given T as

$$P(O(t)|T, \theta_{bg}) = \prod_{p\in\mathcal{P}} (\lambda_p)^{n_p} e^{-\lambda_p T},$$

where $n_p = |\{t_i \in N_p | t_i \in (0, T]\}|$.

For the inhomogeneous Poisson process, the rate parameter $\lambda_p(t)$ is assumed to be a continuous function of time; however, we will consider approximating this as a piecewise constant function over D uniformly spaced divisions in $(0, T]$, with the inhomogeneous rate parameters for phone p denoted $\lambda_{p,d}$ for $d = 1, \ldots, D$. We make a corresponding subdivision in our collection of observations N_p into D partitions specified as

$$N_{p,d} \equiv \{t_i \in N_p | t_i \in ((d-1)\Delta T, d\Delta T], i = 1, \cdots, n_{p,d}\},$$

where $\Delta T = T/D$. Thus, the likelihood of the entire collection of points $O(t)$ under the keyword model given T can be expressed

$$P(O(t)|T, \theta_w) = \prod_{p\in\mathcal{P}} \prod_{d=1}^{D} (\lambda_{p,d})^{n_{p,d}} e^{-\lambda_{p,d}\Delta T}. \quad (2)$$

To this point we have assumed a fixed keyword duration T, so we will now describe how keyword duration is incorporated into the model. Underlying our entire approach is the assertion that words are distinguished by a characteristic pattern of phones in time. We now make a further simplifying assumption that this representative pattern is independent of actual keyword duration. In other words, multiple observations of the same keyword scaled to the interval $(0, 1]$ will result in the same pattern and thus can be modeled by the same set of inhomogeneous rate parameters. To incorporate this, we define a new set of points with respect to a normalized time scale as $N'_p = \{t'_i | t'_i = t_i/T, \ \forall t_i \in N_{p,d}\}$ with $O'(t) = \{N'_p\}_{p\in\mathcal{P}}$. After a change of variables, the probability in (2) with $O'(t)$ becomes

$$P(O'(t)|T, \theta_w) = \prod_{p\in\mathcal{P}} \prod_{d=1}^{D} (T\lambda_{p,d})^{n_{p,d}} e^{-\lambda_{p,d}/D}.$$

Our estimates of $P(O'(t)|T, \theta_w)$ and $P(O(t)|T, \theta_{bg})$ are conditioned on the latent variable T, therefore we may compute the detection function (1) on an unknown utterance by integrating over T in

$$d_w(t) = \log\left[\int_0^\infty \frac{P(O'(t)|T, \theta_w) P(T|\theta_w)}{T^{|O(t)|} P(O(t)|T, \theta_{bg})} dT\right].$$

In practice this integral is approximated by a summation over a discrete set $\mathcal{T}$ of candidate durations spaced at even intervals. We estimate $P(T|\theta_w)$ for each $T \in \mathcal{T}$ based upon keyword examples from training. After finding the parameters for θ_w, θ_{bg} and $P(T|\theta_w)$, we can calculate $d_w(t)$ given an observation $O(t)$. A keyword detection occurs whenever $d_w(t)$ exceeds threshold δ_w which may be determined from development data.

3. Experiments

To evaluate our proposed phone event selection techniques, we began by generating GMM and SMLP-based phone posteriors for the TIMIT database and then derived local maxima and filtered events for various thresholds. To compare these sets of events, we computed our mutual information metric over a range of event thresholds δ, and these results are presented in Figure 3. The mutual information for SMLP events exceeds GMM events, and filtered events slightly outperform local maxima events in both cases. The filtering operation necessarily reduces the magnitude of the peaks in filtered trajectories which accounts for the difference in location of the peak mutual information. The SMLP posteriors employed here yield state of the art performance in standard TIMIT phone recognition experiments, so it is not unexpected that they exhibit higher mutual information than GMM posteriors. Finally, the mutual information of the oracle events at 5.16 bits is exactly the entropy of the input distribution.

For comparison, we replicated the toy experiments on TIMIT presented in [4] using local maxima and filtered events for GMM and SMLP posteriors extracted using the thresholds $\{0.22, 0.26, 0.72, 0.81\}$ indicated in Figure 3. Keyword model parameters (θ_w) and duration statistics for each of the 11 words in the TIMIT sa1 training sentences were computed using transcriptions. The background model parameters (θ_{bg}) were derived from 3696 si/sx type sentences because they were more phonetically balanced. As in the previous work, the test set consisted of 1512 sentences from sa1, si and sx test sentences. After applying the model, keyword detections were declared for local maxima of $d_w(t)$ above threshold δ_w, and detections within

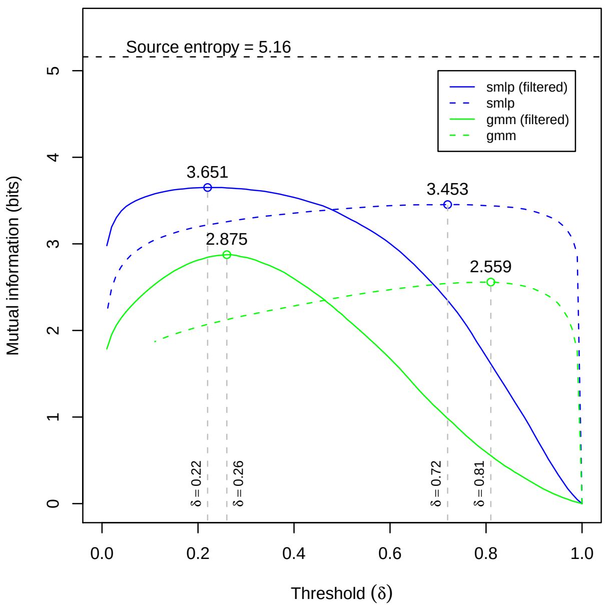

Figure 3: *Comparison of phone event selection techniques as a function of threshold δ for TIMIT si/sx test utterances.*

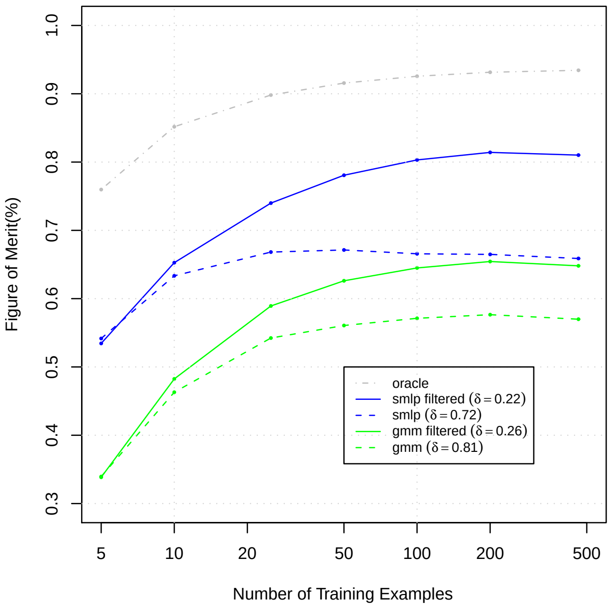

Figure 4: *Average FOM for the keywords in Table 1 as a function of the number of training examples for various types of events with threshold δ indicated.*

Table 1: *Average FOM for various TIMIT sa1 keywords.*

		filtered		local maxima	
keyword	oracle	SMLP	GMM	SMLP	GMM
had	87.2	66.8	54.9	57.1	51.6
dark	98.2	92.2	79.8	67.8	63.6
suit	84.1	67.7	53.9	44.6	38.2
greasy	99.2	96.1	87.6	88.3	89.2
wash	96.4	93.4	85.9	86.3	78.6
water	97.8	77.6	56.9	64.1	40.8
year	91.0	73.3	34.7	52.9	37.1
averages:	93.4	81.0	64.8	65.9	57.0

100 ms of the beginning of the keyword in the transcript were marked as correct. Multiple correct detections of the same keyword were discarded, and all other detections were recorded as false alarms. For the results listed in Table 1, we calculated average figure of merit (FOM) [6], the mean detection rate given $1, 2, \ldots, 10$ false alarms per keyword per hour as the threshold δ_w was varied. In another series of tests, we evaluated FOM as the number of training examples used to generate the keyword model was varied. Average FOM performance for the keywords in Table 1 as a function of the number of training examples is shown in Figure 4.

We observe that the use of filtered events resulted in 23% and 14% relative improvement in average FOM over local maxima for SMLP and GMM, respectively. Examining the mutual information in Figure 3, we observe that that peak mutual information is highly correlated ($\rho = 0.9$) with average FOM in Table 1. Although we have only presented a single threshold common to all phones, we did explore the possibility of optimizing thresholds for each phone and found that it produced negligible ($< 0.2\%$) increases in our mutual information measure. In some instances better FOM results can be obtained by choosing a threshold *lower* than indicated by our metric. For instance using filtered SMLP events, FOM for "greasy" is 98.0 with $\delta = 0.10$ compared to 96.1 with $\delta = 0.22$. While decreasing threshold increases false alarms, we find that the keyword spotting model is much more sensitive to missing true events than false alarms.

4. Conclusions

A very desirable feature of a keyword spotting system is the ability to search through volumes of speech quickly. Regardless of system, faster processing can be achieved by limiting input data to the minimum required. It was previously demonstrated (see [4]) that Poisson process based keyword spotting models operate with performance comparable to traditional HMM-based systems while using a far sparser representation. In this work we have demonstrated the use of phonetic matched filters to produce an even sparser set of events, reducing the event set by 40%, while simultaneously improving average keyword spotting performance by 23%.

5. References

[1] R. C. Rose,"Word spotting from continuous speech utterances," in *Automatic Speech and Speaker Recognition: Advanced Topics*, C. H. Lee and F. K. Soong, and K. K. Paliwal, Eds., Kluwer Academic Publishers, 1996, pp. 303-329.

[2] N. Morgan, H. Bourland, S. Greenberg, H. Hermansky, "Stochastic perceptual auditory-event-based models for speech recognition," in *Proc. of ICSLP*, pp.1943-1946, 1994.

[3] M. Lehtonen, P. Fousek, and H. Hermansky, Hierarchical approach for spotting keywords, *IDIAP Research Report, no.05-41*, 2005.

[4] Jansen, A. and Niyogi, P.,"Point Process Models for Spotting Keywords in Continuous Speech", *IEEE Trans. Audio, Speech and Language Proc.*, 17(8):1457–1470, 2009.

[5] G.S.V.S. Sivaram and H. Hermansky, "Multilayer Perceptron with Sparse Hidden Outputs for Phoneme Recognition," in *Proc. of ICASSP*, 2011.

[6] J.R. Rohlicek, W. Russell, S. Roukos, H. Gish, "Continuous hidden Markov modeling for speaker-independent word spotting" in *Proc. of ICASSP*, pp.627-630, vol.1, 1989.