DETECTION-BASED SPEECH RECOGNITION WITH SPARSE POINT PROCESS MODELS

Aren Jansen

Johns Hopkins University
HLT Center of Excellence
Baltimore, MD USA

Partha Niyogi

The University of Chicago
Depts. of Computer Science and Statistics
Chicago, IL USA

ABSTRACT

We present a bottom-up approach to connected digit recognition in which (i) the speech signal is transformed into a sparse set of acoustic events in time, (ii) point process models (PPM) of these events are used to detect candidate digit occurrences, and (iii) the candidate digit detections are reduced to a single digit sequence prediction by using a previously proposed graph-based optimization. We find the performance of this detection-based system on the AURORA2 evaluation matches that of an HTK baseline in clean speech and provides improved robustness to non-stationary noise. A similar robustness to stationary noise sources is achieved with unsupervised PPM adaptation using small amounts of the noisy data.

Index Terms— speech recognition, speech processing

1. INTRODUCTION

Frame-based speech recognition technologies build statistical dynamic models of vector time series representations that span the entirety of the speech signal. The most common approach uses hidden Markov models to decode linguistic content by finding a sequence of hidden states that best describes each frame of this vector time series. However, when out-of-vocabulary words or non-stationary distortions are introduced, interpreting the entire speech signal with severely mismatched statistical models can introduce insertion errors as well as degrade recognition of uncorrupted regions.

With this motivation in mind, we have conducted several recent investigations into the benefits of temporal sparsity in designing alternative speech recognition representations and models. In [1], we demonstrated that statistical point process models (PPM) of the temporal patterns of sparse acoustic events can be used to detect keyword occurrences in continuous speech with the same accuracy as an equivalent frame-based keyword-filler HMM system. Moreover, we found that sparse PPMs provide a natural resilience to non-stationary babble noise not exhibited by the frame-based system [2]. When combined with a simple unsupervised adaptation strategy, the PPMs exhibited improved robustness more generally.

A compatible detection-based architecture for small and medium vocabulary continuous speech recognition has also been recently proposed [3, 4]. The basic idea is to run in parallel a collection of word detectors and extract a word sequence from their combined output according to some appropriate heuristic. While HMM-based word recognizers perform a Viterbi decode of the entire signal using a network of word and silence models, a detection-based recognizer need only predict word sequences for regions of the speech signal it is reasonably confident about.

In this paper, we combine our PPM techniques with the detection-based speech recognition architecture to construct a connected digit recognizer. The question at hand is whether the past success of temporally sparse modeling will translate to small vocabulary word recognition performance comparable to a modern HMM-based recognizer. To evaluate this, we consider the AURORA2 database, which consists of read connected digit sequences, both in clean form and in the presence of various noise types and levels.

We begin with a brief description of the architecture of the baseline HMM recognizer used in this study. This is followed by complete specification of the PPM recognizer, including the point process representation, Poisson process digit models, graph-based decoding algorithm, and unsupervised noise adaptation procedure. Finally, we present the performance of each system on the AURORA2 evaluation and provide a discussion of the performance trends.

2. BASELINE HMM RECOGNIZER

To provide a frame-based recognition baseline for our experimental results, we consider the standard HMM-based connected digit recognizer architecture specified in the AURORA2 distribution and evaluated in [5]. The acoustic model consists of eleven connected digit models (oh, zero, and one through nine), each consisting of 16 hidden states connected left-to-right with no skip transitions. Taken together with silence model consisting of three states, the entire HMM recognizer contains combined hidden state space $\mathcal{S} = \{s_1, \ldots, s_{179}\}$. Each digit state is modeled with a 3-component Gaussian mixture model (GMM) with diagonal covariance matrices (the silence model states use 6-component GMMs). Evaluation of this baseline architecture was performed using the Hidden Markov Model Toolkit (HTK) v3.4 and a mel frequency cepstral coefficient (MFCC) representation computed by the AURORA Front-End v2.0. These base cepstra were combined with delta and acceleration coefficients for the standard 39-dimensional representation.

3. PPM RECOGNIZER

Our PPM-based small vocabulary architecture consists of the following main components (see Figure 1): (i) an acoustic event detector D_ϕ for each ϕ in some acoustic feature set $\Phi = \{\phi_1, \ldots, \phi_n\}$, (ii) a word detector d_w for each word w in the lexicon $\mathcal{W} = \{w_1, \ldots, w_m\}$, and (iii) a graph-based decoder. Each acoustic feature detector D_ϕ functions to transform the speech signal into a point pattern N_ϕ that consists of the times that acoustic feature ϕ is most strongly expressed. The composite set of point patterns $R = \{N_{\phi_i}\}_{i=1}^n$ defines the sparse point process representation on which all subsequent modeling is based. A probabilistic point process model of each word is used to define a detector d_w that maps R to a collection of candidate intervals of word w, along with their associated confidence scores. Finally, the collected output of

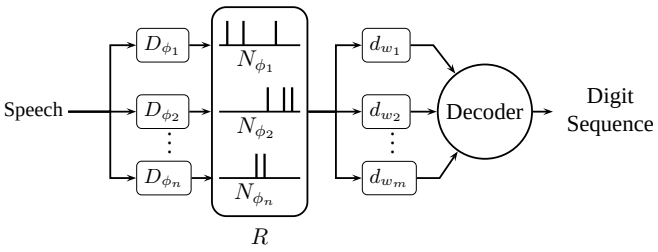

Fig. 1. Architecture of our PPM-based recognizer.

$\{d_{w_j}\}_{j=1}^{m}$ defines the input to the graph-based decoding algorithm of [4], which outputs a single word sequence in $\mathcal{W}^*$.

3.1. Point Process Representation

To provide a controlled comparison to the HTK HMM baseline described above, we used the same MFCC representation and took our acoustic feature set Φ to be the set $\mathcal{S}$ of hidden states of the HMM recognizer (i.e., we constructed one acoustic feature detector from the GMM of each HMM state). As in our previous PPM-based systems, we defined the detector D_s for each state $s \in \mathcal{S}$ as a composition of two operations. First, if $X = \{x_1, \ldots, x_T\}$ is the MFCC vector time series, we compute a feature detector scalar time series $\{g_s(x_1), \ldots, g_s(x_T)\}$ for each state $s \in \mathcal{S}$ according to:

$$g_s(x) = p(s|x) = \frac{p(x|s)}{\sum_{s' \in S} p(x|s')},$$

where $p(x|s)$ is the HTK recognizer's Gaussian mixture model for state s. Next, we apply a thresholded peak finding function to compute the point pattern N_s for each $s \in \mathcal{S}$ as

$$N_s = \{k\Delta | g_s(x_k) > \delta_s \text{ and } g_s(x_k) > g_s(x_{k\pm1})\},$$

where δ_s is the s feature detector threshold and $\Delta = 10$ ms is the sampling interval of X.

Figure 2 shows for an example utterance "926" the HMM emit probability lattice along with the corresponding point process representation, where we have chosen a universal feature detector threshold of $\delta_s = 0.5$ for all $s \in \mathcal{S}$. The frame-based lattice contains 26134 real-valued probabilities (146 frames $\times$ 179 states), while the corresponding point process representation contains in total only 69 real-valued time points. This represents a nearly 400-fold representational data reduction.

3.2. Point Process Digit Detectors

To construct word detectors d_w for each of the digits, we implement a minor variation on the sliding model keyword spotting strategy originally presented in [1]. The basic idea is centered around the assumption that there are two underlying stochastic processes that generate the observed point pattern N_s for each $s \in \mathcal{S}$: a homogeneous Poisson process that generates observations outside instances of a given word and an inhomogeneous Poisson process that generates the point patterns observed within instances of that word.

Each word detector d_w performs two operations, one to compute the confidence that word w occurs at each point in time and a second to compute the most likely duration. We define the detector confidence function f_w at time t as the (log) likelihood ratio

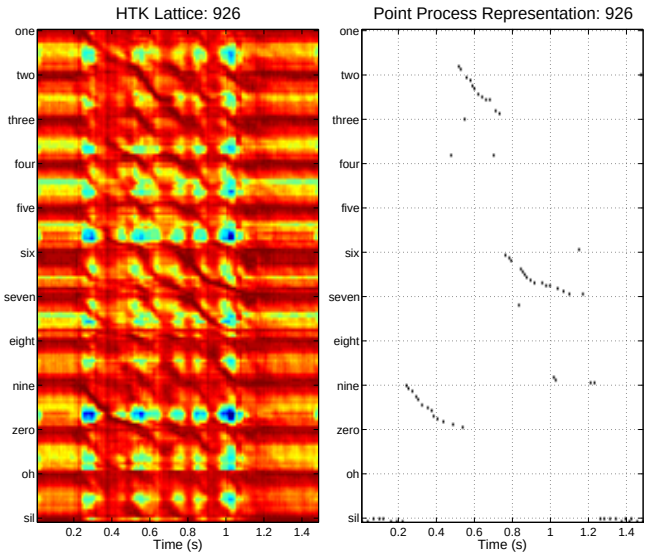

Fig. 2. Log emit probability lattice (left) and corresponding point process representation (right) for an example utterance of "926."

$$f_w(t) = \log \int \frac{P(R_{t,T}|T, \theta_w(t)=1)}{P(R_{t,T}|T, \theta_w(t)=0)} P(T|\theta_w(t)=1) dT, \quad (1)$$

where $\theta_w(t)$ is a hidden indicator function of time that takes the value 1 when the word utterance begins and 0 otherwise, T is a word duration latent variable, and $R_{t,T} = R|_{(t,t+T]}$. The word duration distribution, $P(T|\theta_w(t)=1)$, may be estimated directly from a set of word examples. The remaining two terms are modeled as follows:

1. For the word likelihood $P(R_{t,T}|T, \theta_w(t)=1)$, we begin by normalizing $R_{t,T}$ to the interval $(0, 1]$; that is, we map $R_{t,T}$ to $R'_{t,T}$ such that for each $t_i \in R_{t,T}$ there is a corresponding $t'_i \in R'_{t,T}$ where $t'_i = [t_i - (t-T)]/T$. Assuming that the observations for each instance of the word are generated by a common, T-independent inhomogeneous Poisson process operating on the interval $(0, 1]$ that is subsequently scaled by T to a point pattern on the interval $(t, t+T]$, the likelihood of $R_{t,T}$ takes the form

$$P(R_{t,T}|T, \theta_w(t)=1) = \frac{1}{T^{|R_{t,T}|}} \prod_{s \in \mathcal{S}} e^{-\int_0^1 \lambda_s(u) du} \prod_{u \in N'_s} \lambda_s(u),$$

   where $\lambda_s(u)$ is the process rate parameter at normalized time $u \in (0, 1]$ for feature s and N'_s are the elements of $R'_{t,T}$ for feature s. Learning a new word thus amounts to estimating the Poisson intensity functions, $\{\lambda_s\}_{s \in \mathcal{S}}$, from the point patterns observed when presented with examples of that word. This may be accomplished with kernel smoothing or maximum likelihood estimation of a parametric model.

2. For the background likelihood $P(R_{t,T}|T, \theta_w(t)=0)$, we consider a homogeneous Poisson process model that depends solely on the total number n_s of events observed for each s and the total duration T of the segment. The likelihood of $R_{t,T}$ then takes the form

$$P(R_{t,T}|T, \theta_w(t)=0) = \prod_{s \in \mathcal{S}} [\mu_s]^{n_s} e^{-\mu_s T},$$

where μ_s is the background process rate parameter for feature detector s. Training this model amounts to estimating the rate parameters $\{\mu_s\}_{s\in\mathcal{S}}$ as the average feature detector firing rates over a collection of arbitrary speech not containing the word.

Given a novel utterance, we may evaluate the detector confidence function by sliding a set of windows with durations distributed across the support of $P(T|\theta_w(t)=1)$ and approximating the integral expression of Eq. 1 with a sum. The points in time at which $f_w(t)$ exceeds a confidence threshold γ define the set of candidate word occurrence times. The corresponding candidate word *intervals* are subsequently determined by finding the durations that maximize the integrand of Eq. 1 at each candidate word occurrence.

3.3. Graph-Based Decoder

Now that we have a method of producing a set of candidate digit intervals, it remains to reduce what may be several competing hypotheses to a single digit sequence prediction. The naive approach would be to sort the candidate detections and read off their lexical identity as the predicted sequence. However, the set of digit detectors defined above are prone to making false alarm errors, especially when presented with noisy data. Moreover, each detector can produce a cluster of degenerate detections (i.e. multiple detections by the same detector of a single digit instance), each with a different predicted duration. Given these types of detector errors, such a simple decoding strategy would result in high recognizer insertion rates.

To address the false alarm problem, we factor in the overlap and confidence scores of candidate digit detections by adopting the hypothesis combination approach of [4]. Our implementation consists of two steps:

1. Construct a weighted, directed acyclic graph consisting of one start vertex at time 0, one end vertex at time ∞, and one vertex for both endpoints of each candidate digit detection interval. The edge set is constructed by (i) connecting each vertex to the left boundary vertex of the next (in time) candidate detection with an edges of weight 0; (ii) connecting the left boundary vertex of each candidate detection to its right boundary vertex with an edge of weight $-f_w(t)$; and (iii) connecting the right boundary vertex of each candidate detection to any left boundary vertex that occurred within 20 ms prior with a reversal edge of weight 0 (taking care not to introduce a cycle).
2. Find the min-cost path from the start node to end node using Dijkstra's algorithm (note that since the above-defined graph is acyclic, our negative edge weights do not preclude application). The predicted digit sequence is then simply determined by the order and identity of those candidate detection intervals traversed in the min-cost path.

Like the point process digit models, this sequence decoding algorithm also operates on a sparse, event-based representation (word boundaries in this case). Thus, the PPM recognizer maintains temporal sparsity at each stage of its bottom-up processing.

3.4. Feature Detector Threshold Adaptation

In addition to the core PPM recognizer described above, we also consider the performance of the PPM adaptation strategy, originally presented in [2] for our keyword spotting system. The basic strategy is to adjust acoustic feature detector thresholds to maintain the background firing rates measured in clean speech. Formally, let δ_s be the firing threshold for feature detector D_s and let μ_s be the corresponding background firing rate measured in clean speech. Given some amount of speech in the noisy environment, we can measure the new background firing rate $\mu'_s(\delta'_s)$ as a function of a new phone detector threshold value δ'_s. Noise adaptation is subsequently accomplished by finding the value of δ'_s that produces a rate μ'_s in noisy speech that is closest to the original value μ_s. Since this procedure adapts the feature detector thresholds to maintain clean speech firing rates, we simply reapply the original keyword and background models in the noisy environment.

4. EXPERIMENTS

The AURORA2 database consists of TI-DIGITS training and testing data (downsampled to 8kHz) in clean form and in the presence of eight different types of additive noise (see Table 1) at 20, 15, 10, 5, 0, and -5 dB SNR. We consider the clean training evaluation [5], where the acoustic model is trained using clean training data only and tested in the various noise conditions. Each Poisson intensity function $\lambda_s(t)$ was modeled with a piecewise constant function with 20 equal-length segments. The HTK recognizer was used to provide an automatic segmentation of the training data, from which we extracted our digit examples for PPM training. We set feature detector thresholds to $\delta_s = 0.5$ for all $s \in \mathcal{S}$ and set the word detector confidence threshold to $\gamma = 10$. For our PPM adaptation experiments, we provide approximately one minute of noisy adaptation data randomly selected from the test set for each noise condition.

Table 1 lists the word recognition accuracy, defined as the correctness rate minus insertion rate, for the HMM and PPM (both adapted and not) recognizers in each test condition. For each noise type, the average accuracy across 0-20 dB is listed (as per convention of [5]), as is the average across noise types at each SNR level. The bottom-right value in each subtable is the overall accuracy for each system. There are several trends apparent in these results:

1. When tested on clean speech, our PPM-based system comes only 0.7% (absolute) of matching the word recognition accuracy of the baseline HMM recognizer. Given the extreme sparsity of our point process representation (see Figure 2), we view this near matching of performance as a significant finding. In particular, this result clearly demonstrates that, given a suitable statistical model, the vast majority of the data contained in a frame-based representation can be discarded without significant penalty.
2. The non-adapted PPM recognizer is significantly more robust than the HMM system to non-stationary babble, restaurant, and airport noise at all levels. This is consistent with our keyword spotting robustness findings in [2]. As in that study, we believe this behavior results from the fact that non-stationary noise corrupts the speech signal non-uniformly in time. Thus, if the surges of distortion do not coincide with feature detector firings, temporally sparse models can remain more invariant than frame-based models that account for the entire signal.
3. The non-adapted PPM is less robust than the HMM system to stationary subway, car, exhibition hall, and street noise. When stationary noise is introduced, the probability mass in the state posterior distribution $p(s|x)$ gets shifted in some undefined manner. Thus, even if the relative local maxima structure of each feature detector time series $\{g_s(x_1), \ldots, g_s(\mathbf{x}_T)\}$ remains the same, using a fixed (absolute) feature detector threshold may produce a significant change in N_s. Feature detector threshold adaptation was designed to address this.
4. The adapted PPM system, provided with only one minute of adaptation data, significantly outperforms the baseline HTK rec-

Table 1. Digit recognition accuracies (in %) of the HMM, PPM, and adapted PPM systems on the AURORA2 evaluation.

HTK 3.4 Performance

	Subway	*Babble*	*Car*	*Exhibition*	*Restaurant*	*Street*	*Airport*	*Station*	Average
clean	98.9	99.0	99.0	99.2	98.9	99.0	99.0	99.2	99.0
20 dB	97.1	90.2	97.4	96.4	90.0	95.7	90.6	94.7	94.0
15 dB	93.5	73.8	90.0	92.0	76.2	88.5	77.0	83.7	84.3
10 dB	78.7	49.4	67.0	75.7	54.8	67.1	53.9	60.3	63.4
5 dB	52.2	26.8	34.1	44.8	31.0	38.5	30.3	27.9	35.7
0 dB	26.0	9.3	14.5	18.5	11.0	17.8	14.4	11.6	15.4
-5 dB	11.2	1.6	9.4	9.6	3.5	10.5	8.2	8.5	7.8
Average	69.5	49.9	60.6	65.5	52.6	61.5	53.2	55.6	58.6

PPM Performance

	Subway	*Babble*	*Car*	*Exhibition*	*Restaurant*	*Street*	*Airport*	*Station*	Average
clean	98.4	98.3	98.1	98.5	98.4	98.3	98.1	98.5	98.3
20 dB	94.1	92.2	94.6	92.3	92.7	94.6	92.6	92.7	93.2
15 dB	86.0	83.1	85.9	86.9	84.0	86.5	84.3	83.0	85.0
10 dB	67.4	65.0	63.0	69.1	67.6	68.8	68.8	64.3	66.8
5 dB	40.4	43.8	33.3	37.8	46.4	42.8	46.0	39.8	41.3
0 dB	18.8	22.3	13.6	16.5	23.7	21.8	24.1	17.6	19.8
-5 dB	8.5	10.0	6.5	6.1	10.5	10.1	11.5	9.2	9.1
Average	61.3	61.3	58.1	60.5	62.9	62.9	63.2	59.5	61.2

Adapted PPM Performance

	Subway	*Babble*	*Car*	*Exhibition*	*Restaurant*	*Street*	*Airport*	*Station*	Average
20 dB	94.6	93.6	95.1	94.5	93.7	93.5	94.3	94.6	94.2
15 dB	89.6	89.7	89.9	89.6	90.8	88.6	90.4	90.9	89.9
10 dB	79.4	80.2	76.8	78.0	81.5	78.1	82.2	81.6	79.7
5 dB	61.3	62.5	57.1	53.7	64.3	57.9	66.7	61.0	60.6
0 dB	34.6	35.8	29.0	26.8	38.1	31.8	40.6	35.4	34.0
-5 dB	15.6	16.4	12.6	9.9	17.3	13.8	20.0	15.9	15.2
Average	71.9	72.4	69.6	68.5	73.7	70.0	74.8	72.7	71.7

ognizer in the vast majority of noise types and levels. It is important to emphasize that feature detector threshold adaptation is both fully unsupervised and computationally lightweight, and requires very little noisy data for adaptation (only 1 minute used in our study). Furthermore, both our point process model framework and the associated adaptation strategy are compatible with previously proposed noise-robust front ends (e.g. [6]) and GMM adaptation strategies (e.g. [7]) that have demonstrated success on the AURORA2 evaluation. We leave studying combined performance of our system with other noise robustness techniques for future research.

Finally, note that the PPM-based word detector confidence of Eq. 1 is a function only of the point process arrivals that occur. Thus, if the temporal sparsity of each feature detector is high and/or the individual feature detectors are generally mutually exclusive, PPM decoding can exhibit improved computational efficiency over a frame-based Viterbi search.

5. CONCLUSIONS

We have presented the results of the AURORA2 evaluation on a new approach to connected digit recognition that combines point process word modeling with a detection-based speech recognition architecture. We demonstrated that this method matches the performance of a modern HMM-based recognizer in clean speech. We found that point process modeling leads to significantly improved robustness to non-stationary noise sources. Finally, when provided a small amount of untranscribed adaptation data, our feature detector threshold adaptation scheme provides significant improvements in robustness to stationary noise sources as well.

6. REFERENCES

[1] A. Jansen and P. Niyogi, "Point process models for spotting keywords in continuous speech," *IEEE Trans. Audio Speech Lang. Process.*, vol. 17, pp. 1457–1470, 2009.

[2] A. Jansen and P. Niyogi, "Robust keyword spotting with rapidly adapting point process models," in *Proc. of Interspeech*, 2009.

[3] P. Fousek and H. Hermansky, "Towards ASR based on hierarchical posterior-based keyword recognition," in *Proc. of ICASSP*, 2006.

[4] C. Ma, Y. Tsao, and C.-H. Lee, "A study on detection based automatic speech recognition," in *Proc. of Interspeech*, 2006.

[5] D. Pearce and H.-G. Hirsch, "The AURORA experimental framework for the performance evaluation of speech recognition systems under noisy conditions," in *Proc. of ICSLP*, 2000.

[6] Qifeng Zhu, Markus Iseli, Xiaodong Cui, and Abeer Alwan, "Noise robust feature extraction for ASR using the Aurora 2 database," in *Proc. of Eurospeech*, 2001.

[7] Jinyu Li, Li Deng, Dong Yu, Yifan Gong, and Alex Acero, "HMM adaptation using a phase-sensitive acoustic distortion model for environment-robust speech recognition," in *Proc. of ICASSP*, 2008.