Low Resource Point Process Models for Keyword Spotting Using Unsupervised Online Learning

Samik Sadhu
Electrical Engineering Department
Indian Institute of Science
Bangalore 560012, India
Email: samiksadhu@ee.iisc.ernet.in

Prasanta Kumar Ghosh
Electrical Engineering Department
Indian Institute of Science
Bangalore 560012, India
Email: prasantg@ee.iisc.ernet.in

***Abstract*—Point Process Models (PPM) have been widely used for keyword spotting applications. Training these models typically requires a considerable number of keyword examples. In this work, we consider a scenario where very few keyword examples are available for training. The availability of a limited number of training examples results in a PPM with poorly learnt parameters. We propose an unsupervised online learning algorithm that starts from a poor PPM model and updates the PPM parameters using newly detected samples of the keyword in a corpus under consideration and uses the updated model for further keyword detection. We test our algorithm on eight keywords taken from the TIMIT database, the training set of which, on average, has 469 samples of each keyword. With an initial set of only five samples of a keyword (corresponds to $\sim 1\%$ of the total number of samples) followed by the proposed online parameter updating throughout the entire TIMIT train set, the performance on the TIMIT test set using the final model is found to be comparable to that of a PPM trained with all the samples of the respective keyword available from the entire TIMIT train set.**

I. Introduction

Keyword Spotting (KWS) using Point Process Models (PPM) performs poorly when trained with limited number of training samples [1]. This degradation is detrimental for a situation where not many training samples for the keyword are available, but a good keyword spotting performance is in demand. An example scenario can be the one where lots of intercepted voice communications from a secretive group need to be searched for keywords with very few voice examples. Another application would be to detect keywords in languages with limited linguistic resources, because typical automatic speech recognition (ASR) systems do not support more than 50-100 languages [2]. The idea behind the present work is to initiate a PPM with few available keyword samples and then use carefully chosen newly detected samples using this initial PPM to update the PPM model parameters. Hence, the proposed approach is, in principle, unsupervised in nature and works with a small set of annotated keywords. While there are several unsupervised approaches to KWS [3][2], to the best of our knowledge, there is no work that incorporates an online model updating scheme to enhance the performance over the course of an online learning corpus. KWS experiments with eight keywords from the TIMIT corpus show that a PPM, initialized with only five samples and updated using the proposed online learning algorithm performs as good as a PPM trained with all annotated samples in the online learning corpus (approximately 469 samples, on average, per keyword). We begin with a brief description of PPM for KWS.

II. Point Process Models for Keyword Spotting

PPM for KWS was introduced [1] as a landmark based approach. The algorithms works by detecting phonetic events in the posteriorgram obtained from a Deep Neural Network (DNN), which is trained to map the feature space $\mathcal{F}$ to a probability distribution over a set $\mathcal{P} = \{1, 2, \dots N\}$ where each element corresponds to a phoneme among a set of N phonemes. A phonetic event for a phoneme $p \in \mathcal{P}$ is obtained in a frame where the value of the posteriorgram trajectory (probability of occurrence of that phoneme as a function of time) for that phoneme exceeds a threshold δ. For a given time interval $[t_\alpha, t_\beta]$ of duration $T = t_\beta - t_\alpha$, suppose there are n_p phonetic events for phoneme $p \in P$ at locations $t_\alpha \leq t_1^p < t_2^p < \dots < t_{n_p}^p \leq t_\beta$. Then, a complete observation over the duration T is denoted by

$$O_T = \{N_p | p \in P\} \tag{1}$$

where $N_p = \{t_1^p, t_2^p, \dots t_{n_p}^p\}$.

For a given keyword w, consider a set of K observations denoted by $\mathcal{O}_M^{(w)} = \{O_{T^{(1)}}, O_{T^{(2)}} \dots O_{T^{(M)}}\}$, where $T_w = \{T^{(1)}, T^{(2)}, \dots T^{(M)}\}$ denotes the set of the respective word durations. The observations of phonetic events can be modeled by a point process with piece-wise constant rate parameters [1]. The parameter set θ_w for the model corresponding to keyword w is obtained by maximum likelihood (ML) estimation. The likelihood function of the observations $\mathcal{O}_M^{(w)}$ is obtained as

$$\begin{aligned} P(\mathcal{O}_M^{(w)} | T_w, \theta_w) &= \prod_{m=1}^{M} P(O_{T^{(m)}} | T^{(m)}, \theta_w) \\ &= \prod_{m=1}^{M} \prod_{d=1}^{D} \prod_{p \in \mathcal{P}} (\lambda_{p,d})^{n_{p,d}^{(m)}} \exp\left(\frac{-\lambda_{p,d} T^{(m)}}{D}\right) \end{aligned} \tag{2}$$

where, $\lambda_{p,d}$ is the piece-wise constant approximated rate parameter for phoneme p in d^{th} segment among D constant duration segments and $n_{p,d}^{(m)}$ are the respective counts of the phonetic events in the m^{th} sample of the keyword. Thus,

$\theta_w = \{\lambda_{p,d} | p \in \mathcal{P}, 1 \leq d \leq D\}$. The PPM formulation also requires a background model characterizing the rate of occurrences of each phoneme $p \in \mathcal{P}$ outside the locations of the keywords. Let θ_{bg} be the parameter set for the background model. Then likelihood of an observation O_T given the background model $\theta_{bg} = \{\mu_1, \mu_2, \ldots \mu_N\}$, is given by

$$P(O_T|T, \theta_{bg}) = \prod_{p \in \mathcal{P}} (\mu_p)^{(n_p)} \exp(-\mu_p T) \tag{3}$$

where n_p is the number of occurrences of events of phoneme $p \in \mathcal{P}$ in the interval T and μ_p is the respective rate parameter. The detection function is given by

$$d_w(t) = log\left[\int_0^\infty \frac{P(O_T(t)|T, \theta_w)P(T|\theta_w)}{T^{|O_T(t)|}P(O_T(t)|T, \theta_{bg})} dT\right] \tag{4}$$

where $O_T(t)$ is the observation set as defined in equation (1) over $[t-T, t]$ and $P(T|\theta_w) = P(T|w)$ is obtained from the word duration model $\beta(T|w)$. We model the random variable T by a Gaussian distribution. Thus $\beta(T|w) = \mathcal{N}(T|\mu_w, \sigma_w^2)$, where μ_w and σ_w are estimated by ML criterion for a given w. A keyword w is declared to have occurred if $d_w(t)$ crosses a threshold value γ. There have been a few improvements on the basic PPM algorithm including faster decoding techniques [4] and better event selection [5], use of context-dependent phonemes [6], text-to-speech inspired duration modeling [7] and spoken term detection (STD) using PPM [8]. However, in this work, we use the original PPM algorithm as described by Jansen et al. [1].

III. Unsupervised Online Learning in PPM based KWS

The steps of the proposed unsupervised online learning algorithm are illustrated in Fig. 1. At the beginning of the algorithm, we initialize a PPM with parameters $\theta_w^{(K_{start})}$ learnt from K_{start} training samples of a keyword as described in section III-A and an initial estimate of keyword detection threshold $\gamma(K_{start})$. At any point of the online learning, we use the current model to determine the detector plot $d_w(t)$ on the speech from an online learning corpus as described in the section II (indicated by [A] in Fig. 1). Given the speech, the proposed algorithm detects new occurrences of the keyword using a procedure outlined in section III-B (indicated by [B] in Fig. 1). Once the k^{th} $(k > K_{start})$ sample of the keyword is detected, we update the threshold value to $\gamma(k)$ as described in section III-C and the learning factor to $\alpha(k)$ as described in section III-D (indicated by [C] and [D] in Fig. 1 respectively). The PPM model is updated using the new value $\alpha(k)$ (indicated by [E] in Fig. 1). If no keyword is detected in the given speech, the PPM does not undergo any update.

A. Initial Model

We assume a scenario where not many annotated training samples of the keyword are available. Suppose only K_{start} training samples of the keyword are available to begin with and we train a PPM and a word duration model with these K_{start} samples following the steps outlined in section II. The parameters from the resulting model are used as the initial estimate for the proposed online learning algorithm. However, the word duration model is not updated and remains fixed throughout the learning process.

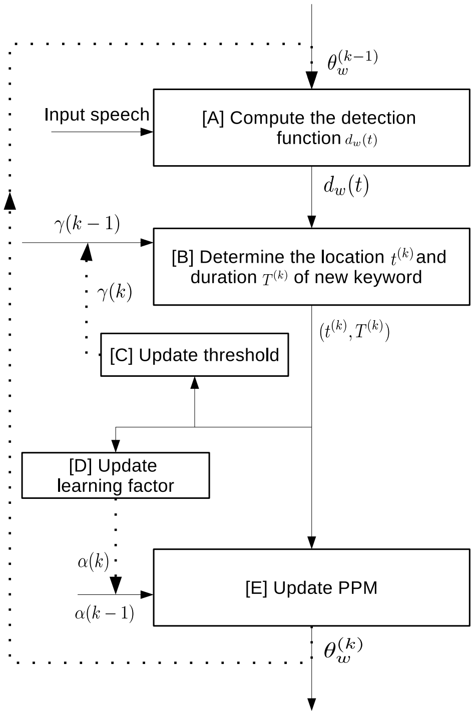

Fig. 1: Block diagram summarizing the online learning steps for updating the PPM.

B. New location and duration hypotheses

Given the speech from the online learning corpus, we obtain the detector plot $d_w(t)$ using equation (4). We estimate the location and duration of a new keyword sample by the following steps:

1) Location $t^{(k)}$ of the k^{th} keyword: Let $\gamma(k-1)$ be the threshold after $(k-1)$ keywords have been detected. Suppose the k^{th} sample is detected in the region $[\tau_1^{(k)}, \tau_2^{(k)}]$ where $d_w(\tau_1^{(k)} + \epsilon) > \gamma(k-1)$ and $d_w(\tau_1^{(k)} - \epsilon) < \gamma(k-1)$ for a small value $\epsilon > 0$ and $\tau_2^{(k)}$ is the first time instant after $\tau_1^{(k)}$ where $d_w(\tau_2^{(k)} + \epsilon) < \gamma(k-1)$ and $d_w(\tau_2^{(k)} - \epsilon) > \gamma(k-1)$ for a small value $\epsilon > 0$. The end location of the k^{th} keyword is hypothesized to occur at

$$t^{(k)} = \arg \max_{\tau_1^{(k)} < t < \tau_2^{(k)}} d_w(t). \tag{5}$$

We also define

$$\beta_k = \max_{\tau_1^{(k)} < t < \tau_2^{(k)}} d_w(t). \tag{6}$$

2) Duration $T^{(k)}$ of the k^{th} keyword occurring at time $t^{(k)}$: From the word duration model $\beta(T|w)$, we consider four potential durations of the keyword and the duration of the k^{th} newly detected sample is estimated using equation (7).

$$T^{(k)} = \arg\max_{n\in\{-1,0,1,2\}} P\left(O_{\mu_w+n\sigma_w}(t^{(k)})\,\middle|\,\mu_w+n\sigma_w,\theta_w^{(k-1)}\right) \times \beta(\mu_w+n\sigma_w|w) \tag{7}$$

C. Updating $\gamma(k)$ after detection of the k^{th} sample of a keyword

A low $\gamma(k)$ value can potentially give rise to a lot of false alarms, which may, in turn, result in a poor PPM. On the other hand, a very high value of $\gamma(k)$ may result in true rejections leading to a poor model too. In our algorithm, after determining k keyword locations and durations, we derive the set

$$M^{(k)}(w) = \{\beta_{\hat{k}}|1 \le \hat{k} \le k\} \tag{8}$$

We propose that the value of $\gamma(k)$ after detecting the k^{th} sample of a keyword be assigned to

$$\gamma(k) = \begin{cases} 0.1 \times median(M^{(k)}(w)) \text{ for } k = K_{start} \\ 0.5 \times median(M^{(k)}(w)) \text{ for } k > K_{start} \end{cases} \tag{9}$$

The set $M^{(K_{start})}(w)$ consists of the values of β_k corresponding to the initial K_{start} samples. We take 10% of the median value at the beginning of the algorithm to encourage accurate detection of keywords as the initial model might not be rich enough to give a high response at new keyword locations. The purpose of choosing the median instead of mean is to avoid $\gamma(k)$ to be influenced by outlier values in $M^{(k)}(w)$ due to false alarms.

D. Model updating

Once the k^{th} sample of w is detected at location $t^{(k)}$ with duration $T^{(k)}$, we update the parameter set θ_w by the following operations.

1) Calculate rate parameter set $\hat{\theta}_w$ for the new example: Consider the current set of rate-parameters at the end of detecting $k-1$ samples of keyword w

$$\theta_w^{(k-1)} = \left\{\lambda_{p,d}^{(k-1)}\middle| p\in\mathcal{P}, 1\le d\le D\right\} \tag{10}$$

A new set of piece-wise constant rate-parameters is obtained for the newly detected keyword by maximizing the likelihood function as

$$\hat{\theta}_w = \arg\max_{\theta_w} P\left(O_{T^{(k)}}(t^{(k)})|T^{(k)},\theta_w\right) \tag{11}$$

The solution to the above optimization problem is obtained as

$$\hat{\theta}_w = \left\{\hat{\lambda}_{p,d} = \frac{n_{p,d}D}{T^{(k)}}\middle| p\in\mathcal{P}, 1\le d\le D\}\right\} \tag{12}$$

where, $n_{p,d}$ is the number of phonetic events for the p^{th} phoneme in the d^{th} segment of the newly detected keyword.

2) Obtaining the updated parameter set $\theta_w^{(k)}$: The updated set of rate parameters is obtained by a convex combination of the elements from the above two sets using a learning factor $\alpha(k)$.

$$\theta_w^{(k)} = \{\lambda_{p,d}^{(k)} = (\alpha(k))\lambda_{p,d}^{(k-1)} + (1-\alpha(k))\hat{\lambda}_{p,d} \; |\, p\in\mathcal{P}, 1\le d\le D\} \tag{13}$$

The choice of a proper value of $\alpha(k)$ is essential for arriving at a good set of model parameters after the algorithm runs over the entire online learning corpus. We choose the value of $\alpha(k)$ to be

$$\alpha(k) = \frac{\sum_{\hat{k}=1}^{k-1} T^{(\hat{k})}}{\sum_{\hat{k}=1}^{k} T^{(\hat{k})}}. \tag{14}$$

It is easy to show (see Appendix A) that with this choice of $\alpha(k)$, $\theta_w^{(k)}$ becomes the ML solution of the parameters by using all the detected k samples.

IV. Experiments and Results

To evaluate our proposed algorithm, we use eight keywords obtained from the TIMIT [9] SA1 and SA2 sentences, namely **greasy, water, dark, wash, carry, oily, suit, year**. We use the TIMIT training set consisting of 4620 sentences for training as well as the online learning corpus for learning the model parameters for each of these eight keywords using the proposed algorithm. The number of keywords in the TIMIT train and test as a pair are (462,74), (479,75), (473,75), (469,74), (463,75), (470,74), (462,74), (473,79) for eight keywords respectively. Out of these 4620 sentences, five sentences containing a keyword are used to train the initial model PPM_{init}. The remaining 4615 sentences are used as the online learning corpus for updating the model using the proposed online learning approach denoted by $PPM_{online}(\zeta)$ where $\zeta \in \{1, 2, \ldots 4615\}$ denotes the index of sentences seen by the algorithm. Hence, $PPM_{online}(\zeta)$ is the new model updated from the previous model $PPM_{online}(\zeta-1)$ by incorporating the keywords detected in the ζ^{th} sentence. Similarly, we also train a PPM $PPM_{all}(\zeta)$, $\zeta \in \{1, 2, \ldots 4615\}$, using the original keyword locations provided in the word transcriptions upto ζ^{th} sentence available in the TIMIT corpus. We use PPM_{online}^{F} and PPM_{all}^{F} to denote the final models $PPM_{online}(4615)$ and $PPM_{all}(4615)$ respectively. The DNN used to generate the posteriorgrams in our experiments is obtained from the Kaldi [10] TIMIT recipe . The features used as input to the DNN are 40 dimensional feature-space maximum likelihood linear regression (fMLLR) features [11] with a context of five frames on either side.

The performance of the algorithm is quantified by the percentage area under the Receivers Operating Curves (ROC) obtained by running the models $PPM_{online}(\zeta)$ and $PPM_{all}(\zeta)$ for $\zeta \in \{1, 2, \ldots, 4615\}$ on the TIMIT test set consisting of 740 sentences. These 740 sentences comprise of all the sentences of 24 speakers from TIMIT core test set (24x10=240 sentences) as well as all sentences of 50 speakers from the development set used by the Kaldi TIMIT recipe (50x10=500 sentences). If the area under the ROC is A upto a false alarm rate of f, then the percentage area under the ROC curve is given by $PA_{ROC} = \frac{100 \times A}{f}$

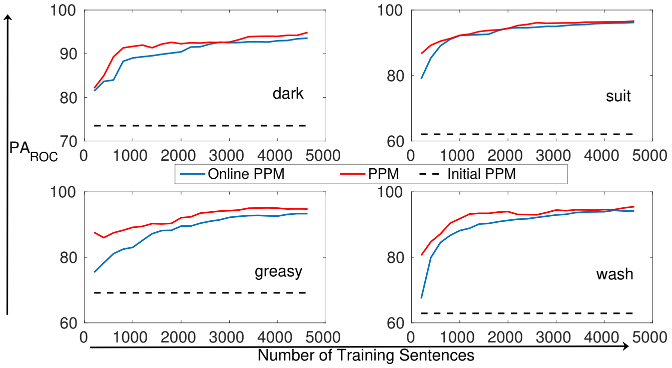

Fig. 2: Variation of PA_{ROC} on TIMIT test set as a function of number of observed sentences in the online learning corpus for the four keywords dark, suit, greasy and wash

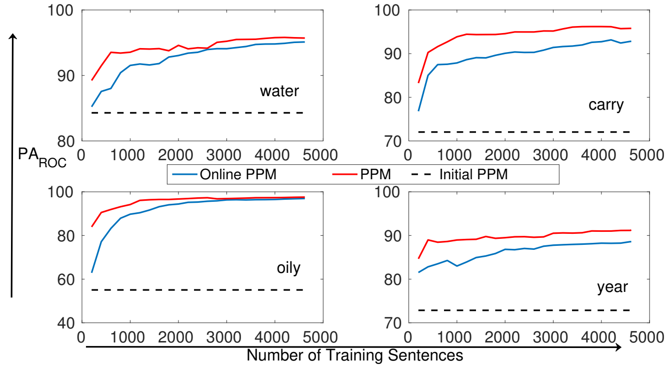

Fig. 3: Variation of PA_{ROC} on TIMIT test set as a function of number of observed sentences in the online learning corpus for the four keywords water, carry, oily and year

The purpose of doing this normalization is to get rid of variations of area under the ROC because of differences in the support of the ROC curves. Another performance measure typically used to evaluate KWS performance is the Figure of Merit (FOM) [12] score, which is the mean of a modified ROC curve sampled at ten points. We have found PA_{ROC} to be better than FOM in capturing gradual improvements of the model because FOM takes the value of the ROC curve at certain number of points and does not quantify the overall change (improvement or deterioration) of the curve. Since each of the models $PPM_{online}(\zeta)$ where $\zeta \in \{1, 2, \ldots 4615\}$, does not change significantly from the previous model, the FOM measure fails to capture the fine change in performance. Hence, we rely on the measure PA_{ROC} to assess the performance of the proposed algorithm. Figs. 2 and 3 show the variation of PA_{ROC} for the models $PPM_{online}(\zeta)$ and $PPM_{all}(\zeta)$ for $\zeta \in \{1, 2, \ldots 4615\}$. It is clear from these figures that the PA_{ROC} from PPM_{online} gets closer to that from PPM_{all} as the ζ increases.

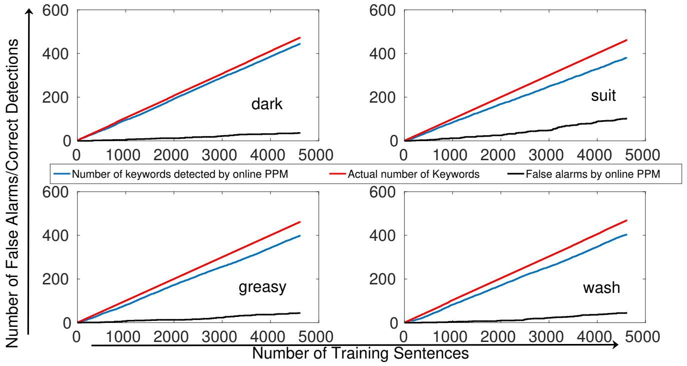

Fig. 4: Number of actual, detected keywords and false alarms by online PPM for dark, suit, greasy and wash on the online learning corpus

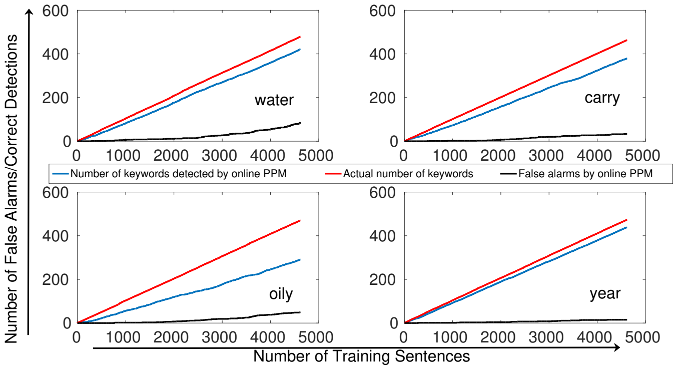

Fig. 5: Number of actual, detected keywords and false alarms by online PPM for water, carry, oily and year on the online learning corpus

On the other hand, Figs. 4 and 5 show a comparison of the actual number of keywords present in the online learning corpus as well as the number of correctly detected keywords and the number of false alarms as a function of ζ. It can be observed that the number of correctly detected keywords as well as the false alarms vary depending on the keyword. This, in turn, determines the quality of the updated PPM. It also suggests that as the number of correctly detected keyword increases, the updated PPM_{online} matches closely with the actual PPM although the online update includes several false alarms.

Table I provides a comparison of the performance of the final model in terms of FOM score. It can be seen that the average FOM scores for the models PPM^F_{online} and PPM^F_{all} on the TIMIT test are comparable and are 3.9% and 4% better respectively than the FOM score of the initial model PPM_{init} on the TIMIT test set. On the other hand, the improvements in PA_{ROC} are 36% and 38% for PPM^F_{online} and PPM^F_{all} respectively over the initial model PPM_{init} which also indicates that the performance of PPM^F_{online} and PPM^F_{all} are similar. This shows that starting with five

ISBN 978-0-9928626-7-1 © EURASIP 2017

Keyword	PPM_{init}	PPM^F_{online}	PPM^F_{all}
dark	94.67	96.93	96.80
suit	89.59	96.23	95.27
greasy	94.59	97.03	97.03
wash	95.81	96.76	97.03
water	96.80	96.67	96.93
carry	92.40	96.93	96.67
oily	85.54	96.49	96.08
year	91.39	92.66	95.06
Average	92.60	96.19	96.36

TABLE I: Comparison of FOM values on TIMIT test set using PPM_{init}, PPM^F_{online} and PPM^F_{all}

examples, the proposed online learning algorithm has updated the parameter set such that it results in a performance similar to that of the parameters obtained from original PPM algorithm [1] which is trained on all the available annotated keywords from the entire TIMIT train database. Hence, the proposed algorithm reduces the required number of training samples to approximately 1% of that required by the original PPM algorithm at the expense of a negligible loss in performance.

V. Conclusions

Using experiments with eight keywords from the TIMIT database we show that the proposed unsupervised online PPM training algorithm gives a comparable performance to the supervised PPM algorithm. This algorithm is useful in scenarios where limited number of annotated keyword samples is available for training, for example, in low-resource languages where the transcription data might not be available, also in situations where the keyword is only used by a secretive group for communication. The key features of the proposed approach is the requirement of as less as 1% of the amount of data required for PPM training to achieve a performance similar to that of PPM. In future work, such unsupervised online learning schemes can be extended to other KWS algorithms and also for unsupervised online training of ASR systems. Although, the proposed algorithm shows promising performance on the TIMIT database, its effectiveness on low resource language databases need to be verified.

Acknowledgment

Financial support for this paper by the Robert Bosch Centre for Cyber Physical Systems at the Indian Institute of Science, Bangalore, and Defence Research & Development Organization, Govt. of India is gratefully acknowledged.

Appendix A
Derivation of $\alpha(k)$

The ML solution for the set $\theta_w^{(k-1)}$ by maximizing the likelihood function (2) with the observations $\mathcal{O}_{(k-1)}^{(w)}$ is obtained as

$$\theta_w^{(k-1)} = \left\{ \lambda_{p,d}^{(k-1)} = \frac{\sum_{\hat{k}=1}^{(k-1)} n_{p,d}^{(\hat{k})} D}{\sum_{\hat{k}=1}^{(k-1)} T^{(\hat{k})}} \middle| p \in \mathcal{P}, 1 \le d \le D \right\} \tag{15}$$

where, $T^{(\hat{k})}$ is the duration of the $\hat{k}^{th}$ detected sample and $n_{p,d}^{(\hat{k})}$ is the number of events for the p^{th} phoneme in the d^{th} segment in the $\hat{k}^{th}$ detected sample. Inclusion of one more training sample to $\mathcal{O}_k^{(w)}$ would modify the solution of the parameter as

$$\begin{aligned}
\theta_w^{(k)} &= \left\{ \lambda_{p,d}^{(k)} = \frac{\sum_{\hat{k}=1}^{k} n_{p,d}^{(\hat{k})} D}{\sum_{\hat{k}=1}^{k} T^{(\hat{k})}} \middle| p \in \mathcal{P}, 1 \le d \le D \right\} \qquad (16) \\
&= \left\{ \lambda_{p,d}^{(k)} = \frac{\sum_{\hat{k}=1}^{k-1} T^{(\hat{k})}}{\sum_{\hat{k}=1}^{k} T^{(\hat{k})}} \frac{\sum_{\hat{k}=1}^{k-1} n_{p,d}^{(\hat{k})} D}{\sum_{\hat{k}=1}^{k-1} T^{(\hat{k})}} + \frac{T^{(k)}}{\sum_{\hat{k}=1}^{k} T^{(\hat{k})}} \frac{n_{p,d}^{(k)} D}{T^{(k)}} \right\} \\
&= \left\{ \lambda_{p,d}^{(k)} = \frac{\sum_{\hat{k}=1}^{k-1} T^{(\hat{k})}}{\sum_{\hat{k}=1}^{k} T^{(\hat{k})}} \lambda_{p,d}^{(k-1)} + \frac{T^{(k)}}{\sum_{\hat{k}=1}^{k} T^{(\hat{k})}} \hat{\lambda}_{p,d} \right\}
\end{aligned}$$

Hence, from equation (16), we can see that choosing $\alpha(k)$ as given is equation (14) ensures that the updated parameter at every detected sample of a keyword exactly matches with the respective ML solution.

References

[1] A. Jansen and P. Niyogi, "Point process models for spotting keywords in continuous speech," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 17, no. 8, pp. 1457–1470, 2009.

[2] Y. Zhang and J. R. Glass, "Unsupervised spoken keyword spotting via segmental DTW on Gaussian posteriorgrams," in *IEEE Workshop on Automatic Speech Recognition & Understanding*. IEEE, 2009, pp. 398–403.

[3] A. S. Park and J. R. Glass, "Unsupervised pattern discovery in speech," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 1, pp. 186–197, 2008.

[4] K. Kintzley, A. Jansen, K. Church, and H. Hermansky, "Inverting the point process model for fast phonetic keyword search." in *INTERSPEECH*, 2012, pp. 2438–2441.

[5] K. Kintzley, A. Jansen, and H. Hermansky, "Event selection from phone posteriorgrams using matched filters." in *INTERSPEECH*, 2011, pp. 1905–1908.

[6] C. Liu, A. Jansen, and S. Khudanpur, "Context-dependent point process models for keyword search and detection-based ASR," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 6025–6029.

[7] K. Kintzley, A. Jansen, and H. Hermansky, "Text-to-speech inspired duration modeling for improved whole-word acoustic models." in *INTERSPEECH*, 2013, pp. 1253–1257.

[8] ——, "Featherweight phonetic keyword search for conversational speech," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 7859–7863.

[9] V. Zue, S. Seneff, and J. Glass, "Speech database development at MIT: TIMIT and beyond," *Speech Communication*, vol. 9, no. 4, pp. 351–356, 1990.

[10] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, "The kaldi speech recognition toolkit," in *IEEE Workshop on Automatic Speech Recognition and Understanding, IEEE Signal Processing Society, December 2011.*

[11] M. J. Gales, "Maximum likelihood linear transformations for HMM-based speech recognition," *Computer speech & language*, vol. 12, no. 2, pp. 75–98, 1998.

[12] J. R. Rohlicek, W. Russell, S. Roukos, and H. Gish, "Continuous hidden Markov modeling for speaker-independent word spotting," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1989, pp. 627–630.