Gaussian Subtraction (GS) Algorithms for Word Spotting in Continuous Speech

Avi Faizakov* Arnon Cohen* Tzur Vaich**

* Electrical & Computer Eng. Dept., Ben-Gurion university, Beer-Sheva, 84105 Israel
** DSP Group, Speech Recognition R&D center, Omer, 84965 Israel

avif(arnon)@ee.bgu.ac.il, tzurv@dsp.co.il

Abstract

In this paper, a novel approach for the design of cohort models for word spotting in continuous speech is presented. This new approach is based on modifying the probability density function of a conventional filler so that regions in the feature space that are related to the keyword will be reduced or removed. By modifying these regions, the filler and keyword models become more orthogonal in the sense that they represent different areas in the feature space, making the filler appropriate to be used as a cohort model. The algorithms, named Gaussian Subtraction (GS) and Gaussian Removal (GR), may be considered discriminative training algorithms.

Introduction

It is well accepted to construct a word spotting system by a *decoder* system followed by a *verifier* as shown in Figure 1 [1, 2]. The decoder system consists of a *Keyword* and *Filler (Garbage)* models. The task of the decoder is to extract, from the continuous speech data, segments "suspected" of containing a keyword, and pass it to the verifier along with a confidence measure. The verifier consists of a *Keyword* model and *Cohorts* anti-models, the task of which is to reject close non-keywords.

One of the problems in the design of Keyword spotting systems is the construction of efficient cohort models. It is a well-known fact in the word spotting literature that the construction of efficient cohort models requires large-scale databases from which the cohorts should be found and trained. Many approaches to the design of cohort models were presented in the literature [1, 2, 3]. Some approaches try to discriminate between the keyword and the cohort models using discriminative-training methods [3]. This paper addresses the problem of creating a cohort-like model for word spotting systems by means of a new GS algorithm. In the GS algorithm suggested in this paper, the cohort-like model is constructed in two stages:

1. A "standard" filler model (a 20-states 300 Gaussians ergodic model is used here) is trained by means of a filler training data.
2. The standard filler model is then adapted to the given keyword by subtracting from it Gaussians that best represent the keyword.

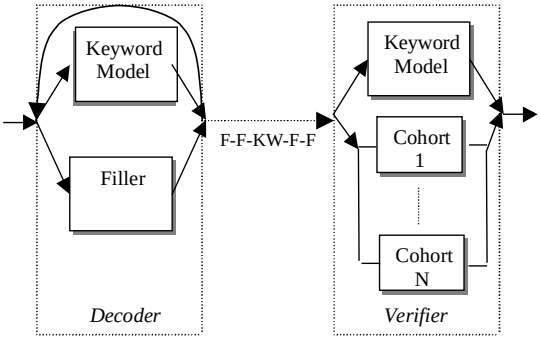

Figure 1: Block diagram of a conventional Word Spotting System

The new filler model represents well all the regions in the feature space besides these of the keyword, but including regions close to them. The GS and GR algorithms are described in the next section. Following, the results without GS and GR are presented, followed by results with GS / GR. Eventually, another verification architecture is described and results for it with and without GR are reported.

GS Algorithm Description

The probability of an observation sequence *O* given a state sequence $q = q_1, q_2, ...q_T$ and a CD-HMM model is given by equation (1) [4]

$$P(O, q \mid \lambda) = P(O \mid q, \lambda) \cdot P(q \mid \lambda) = \pi_{q_1} \cdot b_{q_1}(o_1) \cdot \prod_{t=2}^{T} a_{q_{t-1} q_t} \cdot b_{q_t}(o_t) \quad (1)$$

Where the probability of the jth state at time t is given as a weighted mixture of $M_\lambda(j)$ Gaussians

$$b_j^\lambda(o_t) = \sum_{k=1}^{M_\lambda(j)} c_{jk}^\lambda \cdot \mathrm{N}_{jk}^\lambda(o_t, \mu_{jk}^\lambda, U_{jk}^\lambda) \quad (2)$$

The underlying idea of the GS algorithm is to adapt $b_j^\lambda(\underline{o}_t)$ of each state of the filler model by partially subtracting from it the appropriate keyword model Gaussians. The result is the creation of a "dent" in the

filler model density in the exact location of the keyword. The adapted filler's distribution will thus be

$$b_j^A(o_t) = \sum_{k=1}^{M_F(j)} c_{jk}^F \cdot \mathrm{N}_{jk}^F\left(o_t, \mu_{jk}^F, U_{jk}^F\right) - \\ - \sum_{k=1}^{M_{KW}(j)} \varepsilon_k \cdot \mathrm{N}_{jk}^{KW}\left(o_t, \mu_{jk}^{KW}, U_{jk}^{KW}\right) \quad (3)$$

Where F and KW denote the Filler and Keyword models respectively and $0 \le \varepsilon_k \le 1$ denotes the amount of reduction of the *k*th keyword Gaussian. Note that the adaptation must include normalization so that the integral of the adapted distribution will equal one.

The value of the ε_k's must be determined theoretically or experimentally. The following algorithm performs the adaptation:

- *Gaussians selection*: For each state j, select N_j keyword Gaussians to be subtracted. The *k*th keyword Gaussian is chosen if it is included in the N_j best Gaussians in the sense of having the highest $b_j^F(\mu_k^{KW}), \quad k \in (KW\ Gaussians)$, where b_j^F is the original filler density at filler state j.
- *Mixture Coefficient:* Determine the mixture coefficient of each one of the new Gaussians as $\varepsilon_k = \dfrac{b_j^F\left(\mu_k^{KW}\right)}{N_k^{KW}\left(\mu_k^{KW}\right)}; \quad k \in (N_j\ best)$, where N_k^{KW} is the chosen keyword Gaussian to be subtracted, so that after the subtraction, the new value of the PDF of the filler at the keyword Gaussian mean location will be zero, $b_j^F\left(\mu_k^{KW}\right) - \varepsilon_k \cdot N_k^{KW}\left(\mu_k^{KW}\right) = 0$ (4)
- *Normalization:* Determine normalized weights for the adapted filler model. Define the sum $S_c = \sum_{i=1}^{M_F} c_i - \sum_{k \in N\ best} \varepsilon_k$. For each c_i and ε_k determine $c_i \Leftarrow \dfrac{c_i}{S_c}$ and $\varepsilon_k \Leftarrow \dfrac{\varepsilon_k}{S_c}$ so that their new sum will be 1.

The Gaussian Removal (GR) Algorithm

A simplified approach to the GS algorithm leads to a version of the algorithm called Gaussian Removal (GR) algorithm. The GR algorithm is efficient mainly in SC-HMM. Here we try to make the filler model less probable to the keyword by removing Gaussians. The end result will be a modified filler with reduced number of Gaussians (Note that in the GS approach we have increased the number of Gaussians used by the filler model). The Gaussians to be removed from the filler are chosen as the most probable Gaussians when describing the keyword by the filler model.

This Gaussian removal is done as follows: All the keyword repetitions from the training set are aligned to the filler model, and a Viterbi backtrack is calculated. For each time point (observation vector), the most dominant Gaussian function in $b_j(\underline{o}_t)$ is determined as $k^* = Arg \max_k \left\{c_{jk}^F \cdot \mathrm{N}_{jk}^F\left(o_t, \mu_{jk}^F, U_{jk}^F\right)\right\}$. A histogram of all k^* is calculated. This means that for every one of the filler Gaussians, the number of times it was the most dominant is found. The number N of Gaussians to be removed is determined by the percentage of keyword observations they represent. For example, 10% removal means that the Gaussians removed represented 10% of the keyword observations. The removal begins, naturally, from the most common Gaussian and continues in an orderly fashion.

It should be mentioned that in both the GR and GS algorithms the treatment for all the filler states is the same. This is due to the fact that the model from which we remove or subtract is ergodic, and the temporal meaning of all its states is identical.

Database

The system was evaluated with the Credit Card subset of the switchboard corpus. This speech corpus, originally designed for word spotting tasks, contains 20 words defined as keyword, 9 of which (the ones with the maximal number of repetitions) used in this work. More about the switchboard corpus can be found in [1, 5].

Evaluation

The evaluation of the GS and GR systems was done in terms of their FOM (Average Pd for 0 – 10 FA/Hour) improvement when added as a verification stage to the spotting system as shown in figure 2. In that manner, the efficiency of the models after GS or GR as cohorts is examined.

The decoder stage was first evaluated, namely a CSR-based word spotting system [1] with the architecture shown in the left part of figure 2. Initially, for the nine keywords chosen from the corpus, different number of Gaussians was experimented with in order to select the number of Gaussians that will be used in the removal / subtraction stage. The filler model was a 20-state ergodic model with 300 Gaussians. For the keyword, table 1 summarizes the number of states (in a left to right with one skip architecture, SC - HMM), number of Gaussians, and best FOM achieved from the decoder only network in figure 2 (Left part). It should be mentioned that increasing the number of Gaussians doesn't necessarily give better results, due to the limited amount of training data available. It should be noted that these results are inferior to results presented, for example, in [1]. However, The simplest filler presented in [1] contained 43 phoneme models being used as filler. More complex

systems contained 273 and even 673 models for the entire speech. The system presented here is much simpler in terms of processing time and memory consumption, which were two constraints that guided our work.

Table 1: FOM Results (decoder stage) for the best number of Gaussian functions to represent each keyword (word-wise – SC - HMM).

Keyword	States	Gaussians number	FOM
Credit	18	36	58.40%
card	18	36	34.09%
Interest	18	33	42.60%
Credit Card	22	18	74.22%
Check	14	20	36.13%
dollar	14	25	40.21%
Month	14	45	29.39%
Money	14	30	38.57%
Charge	14	30	66.40%

Next, the complete system of figure 2 was evaluated. In tables 2 and 3, the removal and subtraction results (in terms of FOM) are presented. It can clearly be seen that both algorithms improve the one stage system. It can be seen that the GR has a keyword dependent optimal removal percentage. Figure 3 shows the ROC for the decoder only system, the system with GR verifier, and the system with GS verifier, all for the keyword "Credit Card". An improvement is apparent in low as well as in higher rates of FA/Hour. The GS algorithm also exhibits improvement over the non-adapted case, it is however inferior to the GR results. Work is underway now to improve this algorithm.

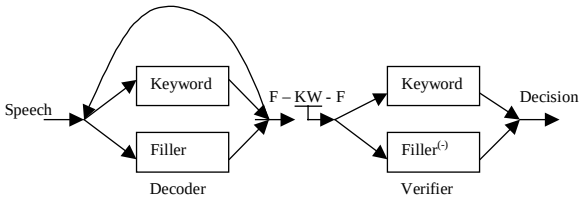

Figure 2: system with GR / GS replacing the cohort model. The (-) in the verifier marks the GS / GR.

Sequential Verifier

So far, the difference between the decoder and the verifier was only the GS and GR algorithms applied in the verifier. In this section, another verification architecture is described.

A histogram analysis of the decodings of the word spotting decoder shows the interesting fact that many times (But not always) the decoded boundaries of the Keyword are different from the manually segmented boundaries estimated by an expert user. The difference is that the manually segmented boundaries are often wider than the decoded ones. To handle that problem, a verification network like the one presented in figure 4 was considered (this network is referred to as "sequential network").

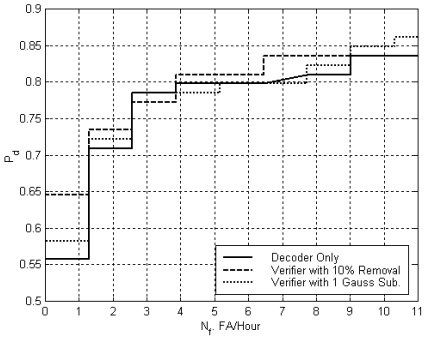

Figure 3: Receiver Operating Curves (ROC) for the keyword "Credit Card".

After the keyword decoded boundaries are found they are artificially widened, based on boundaries misdetection statistics when spotting is performed on the training set. After widening, with high probability the hypothesized word includes several non-keyword frames before and after it. In the network of figure 4, the keyword surrounded by non-keyword frames is better represented due to the filler – keyword – filler path. The advantage of such architecture is that the real keyword location can be found in the contents of the surrounding speech, giving a better description of the verified speech segment. Table 4 presents results with this new architecture for 4 of the words presented above. The improvement is apparent. Figure 5 presents in dashed line the ROC for the keyword "credit" when the sequential network is used. Improvement compared to the initial system can be seen in all FA/Hour rates.

Next, the sequential network with the GR algorithm was evaluated. The keywords "credit", "card", and "credit card" were used with the removal percentage that performed best with the architecture of figure 2. Table 5 presents these results.

Summary and Conclusions

This paper has proposed a new algorithm for the creation of cohort-like model for word spotting systems, to better represent the complementary feature space of a certain keyword. Two versions were presented, the GS algorithm, which may be used with both CD- and SC-HMM and the GR algorithm, which is suited for SC-HMM. The GS and GR algorithms have been applied here only to the filler model, and could in principle be applied also to specially trained cohort models. The proposed approach is easy to implement comparing to discriminative training algorithms, and is more efficient in memory consumption and MIPS terms relative to the conventional system of figure 1.

Table 2: Gaussian removal (GR) results for the system in figure 2, (SC – HMM). Best improvement achieved is highlighted

	Credit	card	Interest	Credit Card	Check	dollar	Month	Money	Charge
0	58.40%	34.09%	42.60%	74.22%	36.13%	40.21%	29.39%	38.57%	66.40%
10	57.47%	34.98%	35.32%	76.29%	37.76%	41.26%	28.18%	38.57%	66.40%
20	59.32%	34.65%	34.03%	72.38%	37.76%	41.26%	30.00%	38.57%	66.40%
50	56.63%	34.42%	34.03%	69.85%	40.33%	46.15%	28.18%	38.57%	66.01%
60	57.32%	34.20%	31.95%	72.04%	44.52%	43.36%	28.79%	38.84%	66.80%
100	42.60%	31.98%	17.92%	74.22%	28.67%	41.96%	31.52%	39.39%	68.77%
Max Improv	0.92%	0.89%	0.00%	2.07%	8.39%	5.94%	2.12%	0.83%	2.37%
								Average	2.62%

Acknowledgments

The authors would like to thank DSP Group (http://www.dspg.com/) for its support in funding this research.

Table 3: Gaussian subtraction (GS) FOM results for the system in figure 2. (SC - HMM)

	Credit	Card	Interest	Credit
0	58.40%	34.09%	42.60%	74.22%
3	58.63%	34.65%	40.52%	75.72%
7	57.55%	34.59%	38.44%	75.95%
15	57.47%	35.03%	34.81%	63.75%
Max	0.23%	0.94%	0.00%	1.73%
			Average	0.72%

Table 4: FOM received from the sequential verifier (without GR) compared with original decoder results.

Word	Original Dec.	Sequential	Improv.
Credit	58.40%	61.17%	2.77%
Card	34.09%	33.92%	-0.17%
Interest	42.60%	49.35%	6.75%
Credit	74.22%	75.72%	1.50%
		Average	2.71%

Table 5: FOM Results for the sequential verifier with GR compared with the sequential verifier without GR and with the verifier of figure 2 (parallel).

Word	Parallel	Seq.	Combina	Improv.
Credit	59.32	61.17	61.86%	0.69%
Credit	76.29	75.72	77.10%	0.81%
Card	34.98	33.92	34.09%	-

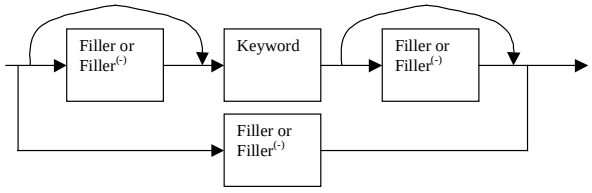

Figure 4: The sequential verifier

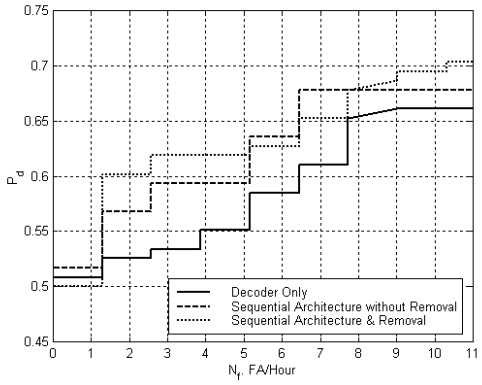

Figure 5: ROC for the word "Credit" with filler – keyword – filler based verifier. Decoder only, sequential verifier, and sequential verifier with GR.

REFERENCES

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

[2] J.R. Rohlicek, "Word spotting," in R.P. Ramachandran and R. Mammone (Eds), *Modern Methods of Speech Processing*, Kluwer Academic Publishers, pp. 123-157, 1995.

[3] R.C. Rose, "Discriminant wordspotting techniques for rejecting non-vocabulary utterances in unconstrained speech", *ICASSP'92*, pp. II-105-8, 1992

[4] L. Rabiner and B. H. Juang, *Fundamentals of speech recognition*, Prentice Hall, 1993.

[5] J. J. Godfrey, E. C. Holliman, and J. McDaniel, "Switchboard: Telephone speech corpus for research and development," *ICASSP'92*, pp. I-517-520.