RESOURCE CONFIGURABLE SPOKEN QUERY DETECTION USING DEEP BOLTZMANN MACHINES

Yaodong Zhang[1], *Ruslan Salakhutdinov*[2], *Hung-An Chang*[1], *James Glass*[1]

[1]MIT Computer Science and Artificial Intelligence Laboratory, Cambridge, Massachusetts, USA
[2]Department of Statistics, University of Toronto, Toronto, Ontario, Canada
ydzhang@csail.mit.edu, rsalakhu@utstat.toronto.edu, {hung_an,glass}@csail.mit.edu

ABSTRACT

In this paper we present a spoken query detection method based on posteriorgrams generated from Deep Boltzmann Machines (DBMs). The proposed method can be deployed in both semi-supervised and unsupervised training scenarios. The DBM-based posteriorgrams were evaluated on a series of keyword spotting tasks using the TIMIT speech corpus. In unsupervised training conditions, the DBM-approach improved upon our previous best unsupervised keyword detection performance using Gaussian mixture model-based posteriorgrams by over 10%. When limited amounts of labeled data were incorporated into training, the DBM-approach required less than one third of the annotated data in order to achieve a comparable performance of a system that used all of the annotated data for training.

Index Terms— spoken query detection, posteriorgram, Deep Boltzmann Machines

1. INTRODUCTION

Spoken query detection can be viewed as a pattern matching problem. If both the spoken query and speech documents use the same representation, finding a query match is equivalent to searching for similar patterns in the documents. A straightforward way of representing both the query and the documents is to convert the speech to text via automatic speech recognition. Detection then becomes a text based search (potentially with confusion networks [1], etc). One of the disadvantages of this approach, however, is poor generalization to arbitrary languages, (or more general audio), since it typically requires a trained speech recognizer. Thus, for under-resourced languages, there is a time/cost issue to obtain enough annotated data to build a recognizer with acceptable recognition performance [2, 3].

In our prior work [4], we have demonstrated an ability to perform spoken query detection without using a speech recognizer. By converting both queries and documents to a posterior probability-based representation called a Gaussian posteriorgram, an efficient lower-bounded Dynamic Time Warping (DTW) algorithm [5] can be used to locate matches in speech documents. The Gaussian posteriorgram is a series of probability vectors computed on frame-based speech features such as MFCCs. Specifically, for each speech frame, a posteriorgram vector is generated by calculating the posterior probability of the MFCCs being generated by each component in a Gaussian mixture model (GMM). The GMM is trained on all MFCCs without requiring any labels.

Although GMM-based posteriorgram produced encouraging results on spoken query detection tasks, we are interested in reducing the performance gap between supervised and unsupervised methods for training posteriorgrams. In this paper we investigate an alternative method for training a posteriorgram representation that is based on Deep Boltzmann Machines (DBMs). We also investigate training the DBM in both unsupervised, and semi-supervised scenarios where a fraction of the training data has been labeled. The DBM is attractive as it has recently been shown to produce good classification results in a variety of domains, including computer vision and information retrieval [6]. The DBM also has the appealing property that it can be trained in a semi-supervised setting.

In this paper we describe the DBM-based posteriorgram representation which we have incorporated into our DTW-based spoken query detection framework. We report the results on several spoken query detection experiments using the TIMIT corpus. In the semi-supervised setting, we observed that 30% of the labeled data are enough to obtain a detection performance that is comparable to the case in which all labeled data are used. In the unsupervised training scenario, we found that a GMM seeded DBM posteriorgram resulted in a 10% relative improvement in equal error rate detection performance over the GMM-posteriorgram baseline.

2. DEEP BOLTZMANN MACHINES

In recent years, deep learning models have been used for phonetic classification and recognition on a variety of speech tasks and showed promising results [7, 8]. A Deep Boltzmann Machine is a network of symmetrically coupled stochastic binary units [6, 9]. It contains a set of visible units $\vec{v} \in \{0,1\}^D$ and a sequence of layers of hidden units $\vec{h}_1 \in \{0,1\}^{F_1}$,

$\vec{h}_2 \in \{0,1\}^{F_2}$,...,$\vec{h}_L \in \{0,1\}^{F_L}$. There are undirected connections only between hidden units in adjacent layers, as well as between the visible units and the hidden units in the first hidden layer (with no within layer connections).

Consider learning a Deep Boltzmann Machine with two hidden layers (i.e. L = 2), the energy of the joint configuration $\{\vec{v}, \vec{h}_1, \vec{h}_2\}$ is defined as:

$$E(\vec{v}, \vec{h}_1, \vec{h}_2; \theta) = -\vec{v}^T W_1 \vec{h}_1 - \vec{h}_1^T W_2 \vec{h}_2 \quad (1)$$

where $\theta = \{W_1, W_2\}$ are the model parameters, representing visible-to-hidden and hidden-to-hidden symmetric interaction terms. (We omit the bias terms for clarity of presentation). The probability of an input vector $\vec{v}$ is given by

$$P(\vec{v}; \theta) = \frac{1}{\mathbb{Z}(\theta)} \sum_{\vec{h}_1, \vec{h}_2} \exp(-E(\vec{v}, \vec{h}_1, \vec{h}_2; \theta)) \quad (2)$$

where $\mathbb{Z}(\theta)$ is the normalization term defined as

$$\mathbb{Z}(\theta) = \sum_{\vec{v}} \sum_{\vec{h}_1, \vec{h}_2} \exp(-E(\vec{v}, \vec{h}_1, \vec{h}_2; \theta)) \quad (3)$$

Exact maximum likelihood learning in this model is intractable, but efficient approximate learning of DBMs can be carried out by using a mean-field inference together with an MCMC based stochastic approximation procedure. Furthermore, the entire model can be efficiently pre-trained one layer at a time using a stack of modified Restricted Boltzmann machines. When modeling real-valued data, we use Gaussian-Bernoulli DBMs. The learning procedure is very similar to the standard binary-binary DBMs (for more details see [6]).

An important property of a DBM is that parameter learning does not require any supervised information. Hierarchical structural information can be automatically extracted as an unsupervised density model to maximize the data likelihood. If any amount of labelling information is given, a standard back-propagation algorithm [10] for multi-layer neural network can be applied to fine-tune the model discriminatively [6]. Furthermore, the back-propagation can be implemented in an online update scheme, hence any future additional labels could be used in online fine-tuning.

3. POSTERIORGRAM GENERATION

In this section, we first review the unsupervised Gaussian posteriorgram generation and then move to semi-supervised and unsupervised DBM based posteriorgram generation.

3.1. Gaussian Posteriorgram

The unsupervised Gaussian posteriorgram is a feature representation of speech frames generated from a GMM. Given a set of N speech frames, let $\vec{x}_1, \ldots, \vec{x}_N$ represent MFCCs for each speech frame. A D-mixture GMM G is trained on all N frames without using any labels. Then, for each speech frame $\vec{x}_i$, a posterior probability, $p_i^j = P(g_j|\vec{x}_i)$, can be calculated where g_j denotes j-th Gaussian component in GMM G. Collecting D posterior probabilities, each speech frame $\vec{x}_i$ is then represented by a probability vector $\vec{p}_i = \{p_i^1, \ldots, p_i^D\}$, where $\sum_j p_i^j = 1 \quad \forall i$.

3.2. Semi-supervised DBM Posteriorgram

Like the phonetic posteriorgrams used in [11, 12], a supervised or semi-supervised DBM posteriorgram is a probability vector representing the posterior probabilities of a set of labeled phonetic units for a speech frame. Formally, if we denotes N speech frames as $\vec{x}_1, \ldots, \vec{x}_N$ and their corresponding phonetic labels $ph_1, \ldots, ph_N$, a posterior probability, $p_i^j = P(ph_j|\vec{x}_i; \theta)$, can then be calculated for any speech frame, $\vec{x}_i$, for each phonetic label ph_j, given DBM model parameters θ and using softmax activation function. If there are V phonetic labels, a speech frame $\vec{x}_i$ can then be represented by a V-dimensional probability vector, $\vec{p}_i = \{p_i^1, \ldots, p_i^V\}$, where $\sum_j p_i^j = 1 \quad \forall i$.

Compared with the Gaussian posteriorgrams which can be generated by a GMM trained without any supervised information, DBM posteriorgrams require some annotated data for training. In the semi-supervised training procedure we use in this work, we first train the DBM model using all data without labels (i.e., unsupervised), followed by the fine-tuning step that requires some amount of labeled data.

3.3. Unsupervised DBM Posteriorgram

In machine learning, a weak classifier can be used to initialize a strong classifier to accelerate the training process. For example, in conventional Expectation-Maximization (EM) training of a GMM, K-means clustering is often used to initialize the target GMM. Inspired by this idea, we investigate a fully unsupervised DBM posteriorgram by training a DBM from labels generated from an unsupervised GMM. Given a set of N speech frames with an MFCC representation, $\vec{x}_1, \ldots, \vec{x}_N$, a D-mixture GMM G is trained on all frames without using any labels. For each frame $\vec{x}_i$, we provide a labeler function L as

$$L(\vec{x}_i) = \arg\max_j P(g_j|\vec{x}_i) \quad (4)$$

where g_j is the j-th Gaussian component in G. In other words, each speech frame is labeled by the index of Gaussian component which maximizes the posterior probability given $\vec{x}_i$. Then a DBM is trained on those "artificial" labels. This DBM posteriorgram generation is similar to the semi-supervised case except that the human produced phonetic label ph_j for each frame is replaced by the GMM produced "artificial" label j. Through this two-stage training process, we leverage the DBM's rich model structure to produce better posteriorgrams than a GMM, while still keeping the entire training framework compatible with the unsupervised setting.

4. SPOKEN QUERY DETECTION

After representing spoken queries and speech documents using posteriorgrams, an efficient DTW algorithm is used to detect possible matches of the query in the documents. The similarity between the keyword query posteriorgram $Q = \{\vec{q}_1, \ldots, \vec{q}_M\}$ with M frames and a speech segment posteriorgram $S = \{\vec{s}_1, \ldots, \vec{s}_N\}$ with N frames is defined by the best warping distortion score as

$$\text{DTW}(Q, S) = \min_{\phi} A_{\phi}(Q, S) \tag{5}$$

where ϕ denotes a particular point-to-point alignment warp and A is the alignment scoring function. The local distance metric used between two frames is an inner product. To accelerate the search efficiency, we developed two lower-bound estimates to help the DTW search [5].

5. EVALUATION

We performed three different evaluations of the DBM-based posteriorgram representation. In the first evaluation, we investigated how different layer configurations of the DBM would affect the quality of the generated posteriorgram as well as the query detection performance. The DBM for this experiment was trained in a fully supervised setting. In the second evaluation, we examined how query detection performance is affected when using partially labeled data for DBM training. In the third evaluation, we compared the query detection performance of the fully unsupervised DBM posteriorgram with our previous Gaussian posteriorgram baseline.

5.1. Spoken Query Detection Task

The spoken query detection task was based on the 630 speaker TIMIT corpus which includes a training set of 3,696 utterances and a test set of 944 utterances. As in [4, 5], 10 query keywords were randomly selected and 10 examples of each keyword were extracted from the training set. For each keyword example, the query detection task was to rank all 944 utterances from the test set based on the utterance's possibility of containing that keyword. Performance was measured by the average equal error rate (EER): the average rate at which the false acceptance rate is equal to the false rejection rate.

5.2. Supervised Results

In the supervised experiments, we used all labeled data (3,696 utterances) in order to maximize the performance while changing different DBM layer configurations. For DBM training, each training utterance was segmented into a series of 25ms windowed frames with a 10ms shift (i.e., centisecond analysis). Each frame was represented by 39 MFCCs stacked with the neighboring 10 frames (5 on each side). In total, the feature dimension for each frame is 429 (39 x 11). All 61 phonetic labels were used for training. After training, each frame in the training and test set was decoded by the DBM, producing a posteriorgram vector of 61 dimensions. Query detection was done by comparing the keyword example posteriorgrams with the test set posteriorgrams using the DTW method described in Section 4.

Table 1. Different DBM configurations.

DBMs	Avg. EER
500	10.6%
300x300	10.3%
500x500	9.8%
1000x1000	10.4%
500x500x500	10.1%

Table 1 presents the results for different DBM configurations and their resulting average EER. In the first column, 500 indicates a DBM that has one layer with 500 hidden units, while 500x500 denotes a DBM with two layers each of which has 500 hidden units. The forward layer training in each configuration was set to stop at the 100^{th} iteration, while the fine-tuning using back-propagation was set to stop at the 50^{th} iteration. The results indicate that detection performance was not overly sensitive to DBMs with different layer settings. This implies that we need not be overly concerned about the DBM layer configurations in subsequent experiments.

5.3. Semi-supervised Results

In the second experiment, we used a two-layer DBM with 500 hidden units for each layer. We first trained our model on all 3,696 unlabeled utterances, followed by the fine-tuning stage that only used partially labeled data. Figure 1 demonstrates the results. On the x-axis, a training ratio of 0.1 indicates that only 10% of the labeled data were used in the fine-tuning stage, while a training ratio of 1.0 means all labeled data were used. It can be observed that the average EER curve drops dramatically from 0.01 to 0.2 and becomes steady between 0.3 to 0.8. This is an interesting result because in scenarios where fully labeled data are not cost effective to obtain, 20% to 30% of labeled data are enough to produce a system that is only slightly worse than the system trained on all labeled data. Moreover, since in the fine-tuning step, the back-propagation algorithm has to go through each data point for each iteration, using a smaller portion of labeled data also saves a significant amount of computing time.

5.4. Unsupervised Results

In the unsupervised training experiment, a 500x500 DBM was trained by using labels generated from a GMM with 61 Gaussian mixtures. Specifically, a GMM was first trained on frame-based MFCCs without using any labels. To be consistent with our prior work, only 13 MFCCs per frame were used to train the GMM. Once the unsupervised GMM had been

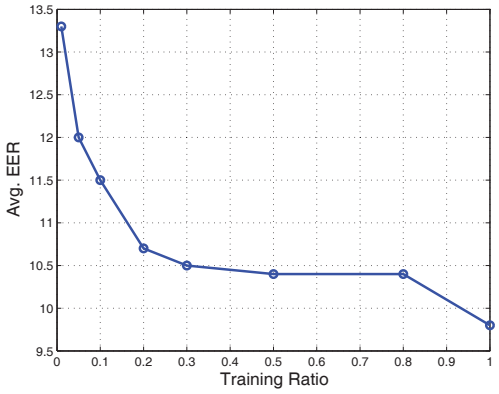

Fig. 1. Average EER against different training ratios

Table 2. Comparison of Gaussian and DBM posteriorgram

Posteriorgram	Avg. EER
Gaussian	16.4%
DBM	14.7%
DBM (1%)	13.3%

created, each frame was subsequently labeled by the most likely GMM component (Eq. 4). A DBM was then trained on 429 MFCCs per frame using the GMM-generated labels. We then compared the unsupervised posteriorgram detection performance between the GMM and the DBM-based posteriorgrams, as shown in Table 2. As we have reported previously [4], the Gaussian posteriorgrams produced an average EER of 16.4%. The unsupervised DBM-based posteriorgrams improved upon this result by over 10% to achieve an average EER of 14.7%. We believe the improvement is due to the DBM's explicit hierarchical model structure that provides a finer-grained posterior representation of potential phonetic units than those that can be obtained by the Gaussian posteriorgram. Note that in an attempt to make a comparison using the same larger feature set, we also trained an unsupervised GMM using the 429 dimensional MFCC vectors that were used to train the DBM. In this case, however, the average EER degraded to over 60%, which we attribute to a weaker ability of the GMM to cope with higher dimensional spaces.

The third row in Table 2, highlights one final advantage of the DBM framework in that it is able to incorporate partially labeled data. When we included only 1% of labeled data, we see that the average EER is further reduced to 13.3% (as also shown in the first data point in Figure 1). This reduction corresponds to another 9.5% performance gain over the unsupervised case.

6. CONCLUSION AND FUTURE WORK

In this paper we presented a spoken query detection method based on posteriorgrams generated from Deep Boltzmann Machines (DBMs). The proposed representation can be easily adapted to work in both semi-supervised and unsupervised training conditions. Spoken query detection experiments on the TIMIT corpus showed a 10.3% relative improvement compared to our previous Gaussian posteriorgram framework in the unsupervised condition. In the semi-supervised setting, the detection performance using the DBM posteriorgram can achieve a comparable performance to fully supervised training when using only 30% of the labeled data.

In future work we plan to perform keyword detection experiments on larger spoken query tasks, and with languages other than English, since the unsupervised DBM posteriorgram DTW framework is language independent.

7. REFERENCES

[1] H. Lin, A. Stupakov, and J. Bilmes, "Improving multi-lattice alignment based spoken keyword spottings," in *Proc. ICASSP*, 2009, pp. 4877–4880.

[2] A. Garcia and H. Gish, "Keyword spotting of arbitrary words using minimal speech resources," in *Proc. ICASSP*, 2006, pp. 949–952.

[3] A. Jansen, K. Church, and H. Hermansky, "Towards spoken term discovery at scale with zero resources," in *Proc. Interspeech*, 2010, pp. 1676–1679.

[4] Y. Zhang and J. Glass, "Unsupervised spoken keyword spotting via segmental DTW on Gaussian posteriorgrams," in *Proc. ASRU*, 2009, pp. 398–403.

[5] Y. Zhang and J. Glass, "An inner-product lower-bound estimate for dynamic time warping," in *Proc. ICASSP*, 2011, pp. 5660–5663.

[6] R. Salakhutdinov, *Learning Deep Generative Models*, Ph.D. thesis, Dept. of Computer Science, University of Toronto, 2009.

[7] A. Mohamed, G. E. Dahl, and G. E. Hinton, "Deep belief networks for phone recognition," in *Proc. NIPS Workshop on Deep Learning for Speech Recognition and Related Applications*, 2009, pp. 1–9.

[8] G. E. Dahl, D. Yu, L. Deng, and A. Acero, "Large vocabulary continuous speech recognition with context-dependent DBN-HMMs," in *Proc. ICASSP*, 2011, pp. 4688–4691.

[9] R. Salakhutdinov and G. E. Hinton, "Deep Boltzmann Machines," in *Proc. of the International Conference on Artificial Intelligence and Statistics*, 2009, vol. 5, pp. 448–455.

[10] G. E. Hinton, S. Osindero, and Y. W. Teh, "A fast learning algorithm for deep belief networks," *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[11] T. Hazen, W. Shen, and C. White, "Query-by-example spoken term detection using phonetic posteriorgram templates," in *Proc. ASRU*, 2009, pp. 421–426.

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