Using a DBN to integrate Sparse Classification and GMM-based ASR

Yang Sun, Jort F. Gemmeke, Bert Cranen, Louis ten Bosch, Lou Boves

Centre for Language and Speech Technology, Radboud University Nijmegen, The Netherlands

{Y.Sun, J.Gemmeke, B.Cranen, L.tenBosch, L.Boves}@let.ru.nl

Abstract

The performance of an HMM-based speech recognizer using MFCCs as input is known to degrade dramatically in noisy conditions. Recently, an exemplar-based noise robust ASR approach, called sparse classification (SC), was introduced. While very successfully at lower SNRs, the performance at high SNRs suffered when compared to HMM-based systems. In this work, we propose to use a Dynamic Bayesian Network (DBN) to implement an HMM-model that uses both MFCCs and phone predictions extracted from the SC system as input. By doing experiments on the AURORA-2 connected digit recognition task, we show that our approach successfully combines the strengths of both systems, resulting in competitive recognition accuracies at both high and low SNRs.

Index Terms: noise robustness, speech recognition, dynamic bayesian network, sparse classification

1. Introduction

The type of speech recognizer that has dominated the speech recognition field for the last 30 years, is undoubtedly an HMM-based recognizer using MFCCs (Mel-frequency Cepstral Coefficients) as input. While quite successful in dealing with clean, read or prepared speech, the performance of this type of recognizer is known to degrade dramatically under noisy conditions or spontaneous conversational speech. Despite the many modifications that have been proposed to different modules of HMM-based ASR systems, a large performance gap still remains between ASR and Human Speech Recognition (HSR) [1, 2]. There is growing consensus that simply not all relevant speech phenomena can be covered in the form of HMMs operating on MFCC's. To close the gap we need novel approaches, perhaps not to completely replace HMM's, but at least to provide additional information beyond that provided by MFCC features modelled by mixtures of Gaussians.

One such an approach that aims to improve the robustness against background noise, *Sparse Classification* (SC), is based on the idea that speech signals can be represented as a sparse linear combination of suitably selected speech segments, *exemplars* [3, 4]. With noisy speech being modelled accordingly as a linear combination of both clean speech and noise exemplars, the model is inherently noise robust when a suitable dictionary of speech and noise exemplars is provided. In SC each speech exemplar, which spans multiple frames to model dependencies between neighbouring frames, is labelled using an HMM-based state-description. Using these labels, the weights of the linear combination of speech exemplars can be converted to likelihoods of the corresponding state labels.

Experiments in [5] showed that SC performs quite well in low SNR cases, but that performance suffers in high SNR conditions. One way of improving SC recognition accuracy at high SNRs while retaining the robustness at low SNRs is by combining the state likelihoods provided by the SC framework with those provided by Gaussian Mixture Models (GMMs) modelling MFCC's. In this paper, we propose to combine these two modelling approaches using a Dynamic Bayesian Network (DBN).

In the last decade, Dynamic Bayesian Networks (DBN) have been introduced as a subset of graphical models that constitute an overarching platform encompassing many existing algorithms for ASR [6]. The DBN framework also allows one to make explicit assumptions about hierarchical relationships between modelling variables that have remained implicit in current models. This greatly facilitates the extension of existing models and, more importantly, the implementation of novel ideas which are difficult to capture with conventional methods [7]. It has been shown that a DBN makes it possible to combine the usual MFCC features in an HMM-based system with estimates of the phones related to feature vectors/frames. For example [8] obtained promising results with adding phone predictors obtained from a noise-robust Bidirectional Long-Short-term Memory Recurrent Neural Network in an in-car alphabet recognition task.

In this work, we use a similar approach, by using the state likelihoods provided by SC to obtain phone predictors as an additional information source in the DBN. By doing experiments on the connected digit recognition task AURORA-2, we investigate to what extent it is beneficial to adapt the relative weights of the SC stream and the MFCC stream depending on the SNR. Accordingly, this paper has two goals. First, we investigate whether a combination of MFCC and SC information can compensate for the accuracy loss of SC at high SNRs. The second goal is to evaluate to what extent different weights of the two information sources can further improve performance across the entire SNR range.

The rest of this paper is organized as follows. In Section 2 we review the fundamental algorithm of sparse classification. In Section 3, we introduce the DBN architecture and explain how SC information is incorporated. We will describe our experiments and discuss the results in Section 4. Finally, we present our conclusions and ideas for future work in Section 5.

2. Sparse Classification

2.1. A sparse representation of noisy speech

In ASR, speech signals are represented by their spectro-temporal distribution of acoustic energy, a *spectrogram*. The magnitude spectrogram describing a clean speech segment $\boldsymbol{S}$ is a $B \times T$ dimensional matrix (with B frequency bands and T time frames). To simplify the notation, the columns of this matrix are stacked into a single vector $\boldsymbol{s}$ of length $D = B \cdot T$.

We assume that an observed speech segment can be expressed as a sparse, linear, non-negative combination of clean speech exemplars $\boldsymbol{a}_j^s$, with $j = 1, \ldots, J$ denoting the exemplar

index. We model noise spectrograms as a linear combination of noise exemplars $\boldsymbol{a}_k^n$, with $k = 1, \ldots, K$ being the noise exemplar index. This leads to representing noisy speech $\boldsymbol{y}$ as a linear combination of both speech and noise exemplars:

$$\boldsymbol{y} \approx \boldsymbol{s} + \boldsymbol{n} \tag{1}$$

$$\approx \sum_{j=1}^{J} \boldsymbol{x}_j^s \boldsymbol{a}_j^s + \sum_{k=1}^{K} x_k^n \boldsymbol{a}_k^n \tag{2}$$

$$= [\boldsymbol{A}_s \boldsymbol{A}_n] [\boldsymbol{x}^s \boldsymbol{x}^n] \tag{3}$$

$$= \boldsymbol{A}\boldsymbol{x} \quad \text{s.t.} \quad \boldsymbol{x}^s, \boldsymbol{x}^n, \boldsymbol{x} \geq 0 \tag{4}$$

with $\boldsymbol{x}^s$ and $\boldsymbol{x}^n$ sparse representations of the underlying speech and noise, respectively. Matrix $\boldsymbol{A}$ has dimensionality $D \times L$, where $L = J + K$.

In order to obtain $\boldsymbol{x}$, we minimize the cost function:

$$d(\boldsymbol{y}, \boldsymbol{A}\boldsymbol{x}) + ||\boldsymbol{\lambda} .* \boldsymbol{x}||_1 \quad \text{s.t.,} \quad \boldsymbol{x} \geq 0 \tag{5}$$

with distance function d and the second term a sparsity inducing L-1 norm of the activation vector weighted by element-wise multiplication (operator $.*$) with vector $\boldsymbol{\lambda} = [\lambda_1 \ \ \lambda_2 \ldots \lambda_L]$. As a distance measure d we use the generalized Kullback-Leibler (KL) divergence. The cost function (5) is minimized using a multiplicative updates routine as in [5].

2.2. Classification using associated state labels

Each exemplar in the speech part of the dictionary $\boldsymbol{A}_s$ is labelled using HMM-state labels obtained from a conventional MFCC-based decoder. Using a frame-by-frame state description of the training data used to construct the dictionary, we associate each exemplar $\boldsymbol{a}_j^s$ with a label matrix $\boldsymbol{\mathcal{L}}_j$, of dimensions $Q \times T$, with Q the total number of states in the system. The matrix $\boldsymbol{\mathcal{L}}$ is a binary matrix containing for each frame $\tau \in [1, T]$ a single nonzero value for the corresponding active state. For each observed speech segment, we now calculate the unscaled likelihood matrix as:

$$\boldsymbol{L} = \sum_{j=1}^{J} \boldsymbol{\mathcal{L}}_j x_j^s \tag{6}$$

As in [5], we increase the likelihood of the silence states by adding a value based on the estimated speech activity in each speech segment. Finally, the likelihoods are normalized to unity for each window.

In order to decode utterances of arbitrary lengths, we adopt a sliding time window approach as in [5]. In this approach, we represent a noisy utterance as a number of fixed-size, overlapping speech segments. For each segment, we calculate a likelihood matrix as described above. Finally, we obtained a likelihood matrix for the entire utterance by averaging the likelihoods of the frames of all the windows that overlap, taking into account the exact temporal positions of the frames.

In [5], decoding was done directly with this *state*-likelihood matrix using Viterbi decoding. In this work, we use *phone*-likelihoods instead in order to reduce the computational complexity. Using a state-to-phone mapping that maps each state to one out of 20 labels (19 different phones plus a silence label), we linearly mapped the state likelihoods to phone likelihoods by summing all the state probabilities underlying each phone. Finally, we further reduce the computational complexity by only retaining the index of the most likely phone at each time frame as our SC observation.

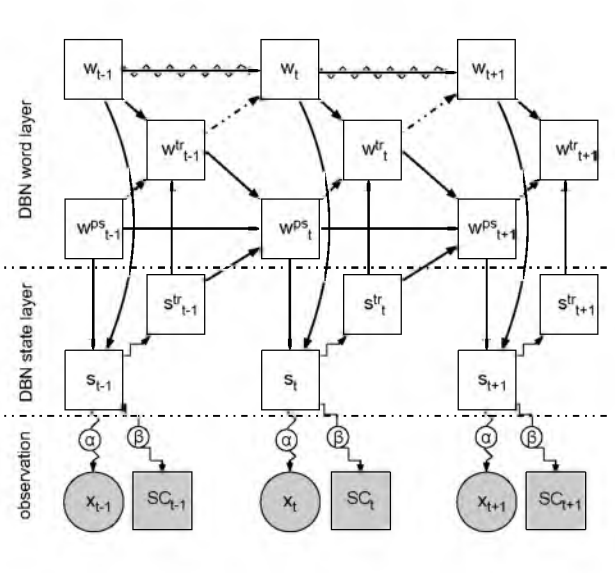

Figure 1: Architecture of the dual-input DBN.

3. Dual-input DBN architecture

Figure 1 depicts the DBN architecture used in this study and is −except for the additional SC input variable− identical to the 'auroraTutorial' structure that comes with the GMTK software distribution [9]. White symbols represent hidden variables while observed variables are shaded; discrete variables are represented by squares and continuous variables are represented by circles. Furthermore, straight lines represent deterministic relationships, while zigzagged lines indicate a probabilistic relation controlled by discrete conditional probability tables (CPT's). Dotted lines correspond to a switching parent dependency.

The hierarchical structure in Figure 1 consists of 3 layers and reflects how in the word layer at each time frame a word is being represented by the discrete variable w_t and how each word in the state layer is assumed to be composed of (a fixed number of) states (represented by the discrete variable s_t). The fixed number of states per word is enforced the use of the variable w_{ps} (cardinality is 16) to keep track of the state position within a word, and the variables s_t^{tr} and w_t^{tr} (both having cardinality 2) to signal any state and word transitions, respectively. Since our vocabulary distinguishes 11 digits ('zero' through 'nine' and 'oh') plus two silence models, the cardinality of the word variable is 13. Using 16 states for each digit, and 3 states for silence (of which one state is shared with a short pause model), the cardinality of the state variable is $11 \times 16 + 3 = 179$. Finally in the observation layer, x_t indicates the traditional MFCC features and SC_t the extra (discrete) feature obtained from the SC system that indicates the index of the most probable phone. The variables α and β ($0 \leq \alpha \leq 1$, and $\beta = 1 - \alpha$) are coefficients to weigh the contribution of the MFCC and SC feature streams respectively.

Denoting the sequence of values that a variable assumes in subsequent frames during the interval $[1, T]$ as $[\cdot]_{1:T}$, the DBN in Figure 1 complies with the following factorization of the joint probability:

$$p(w_{1:T}, w^{tr}_{1:T}, w^{ps}_{1:T}, s^{tr}_{1:T}, s_{1:T}, x_{1:T}, SC_{1:T}) =$$
$$\prod_{t=1}^{T} P(x_t|s_t)^{\alpha} p(SC_t|s_t)^{\beta} f(s_t|w^{ps}_t, w_t) f(w^{tr}_t|w^{ps}_t, w_t, s^{tr}_t)$$
$$p(s^{tr}_t|s_t) f(w^{ps}_1) p(w_1) \prod_{t=2}^{T} p(w_t|w^{tr}_{t-1}, w_{t-1})$$
$$f(w^{ps}_t|s^{tr}_{t-1}, w^{ps}_{t-1}, w^{tr}_{t-1}) \quad (7)$$

where function $p(\cdot)$ represents CPTs with discrete probabilities and $f(\cdot)$ deterministic CPTs only containing zero's and ones. Finally, $P(x_t|s_t)$ is a probability density function which is described by diagonal covariance Gaussian Mixtures.

4. Experiments and Results

4.1. The input features for the DBN

Our DBN system was trained on various combinations of MFCC and SC input features. All input features used in training were obtained from the clean training set of the AURORA-2 corpus (8440 utterances). For testing purposes, we used part of test set 'A', i.e., utterances of four noisy types (subway, car, babble, exhibition hall) at SNR levels 0dB, 10 dB, 20dB and clean. Each subset contains 1001 utterances consisting of a sequence of one to seven digits, 'zero-nine' and 'oh'.

The MFCC input to the DBN consisted of 39 dimensional vectors containing 12 cepstral features plus a separate log-energy coefficient as well as the corresponding first and second order delta coefficients. They were based on a 23 band mel frequency spectrum using a frame shift of 10ms and a frame length of 25ms. Subsequently, the MFCC coefficients were mean and variance normalized. The MFCC feature vectors are represented by diagonal covariance Gaussian Mixtures, which were split once 0.02% convergence was reached. Our final model consists of up to 32 diagonal covariance Gaussian Mixtures.

For deriving the SC information, we used the same configuration as in [5]. A dictionary is created with 4000 noise and 4000 speech exemplars by randomly selecting exemplars from noise and clean speech in the multi-condition training set with the window length T=10. The overlap between two neighbouring windows is constant at 1 frame. The output of the SC system is a 179 dimensional vector for each frame, corresponding to the probability of each state used in the DBN model. In order to reduce the computational load (cf. section 2) we mapped this vector to a single discrete index representing the most likely phoneme for each frame. This phone index derived from the SC is then regarded as a second observation for our system.

4.2. DBN specification

In order to be able to compare the relative contributions of the MFCC and the SC streams to the system as a whole, we used three different combinations of the weights α and β:

1. $\alpha = 1$ and $\beta = 0$
2. $\alpha = 0$ and $\beta = 1$
3. $\alpha = \beta = 0.5$

As can be inferred from Figure 1, case 1 is similar to the traditional system where the MFCCs are the only observations. Likewise, in case 2 the SC feature becomes the only observation. Finally, in case 3 the MFCC and SC streams contribute equally and independently.

4.3. Results

Table 1 shows, besides some previously published results from other studies that can serve as a baseline, the word accuracies for different DBNs on our test set. In order to be able to compare different systems on the basis of a single figure of merit, the bottom row shows the mean recognition rates averaged over all the four noise types (subway, car, babble, exhibition hall) and all used SNR levels.

Table 1: *Word recognition accuracy in %. The 0.95 confidence intervals (assuming a binomial distribution) are printed between brackets.*

	A	*B*	*MFCC*	*SC*	*MFCC/SC*
SNR0	17.1 (1.2)	**67.8 (1.4)**	56.5 (1.5)	60.8 (1.5)	66.8 (1.5)
SNR10	67.3 (1.5)	88.3 (1.0)	90.7 (0.9)	88.1 (1.0)	**93.4 (0.8)**
SNR20	95.0 (0.7)	93.4 (0.8)	97.5 (0.5)	92.8 (0.8)	**97.8 (0.5)**
clean	**98.9 (0.3)**	95.9 (0.6)	98.8 (0.3)	94.6 (0.7)	98.8 (0.3)
mean	69.6 (0.7)	86.3 (0.5)	85.9 (0.5)	84.0 (0.6)	**89.2 (0.5)**

Results copied from [7] and [5] are shown in columns A and B respectively, and serve as a baseline to compare our results with. Column A corresponds to a traditional HMM recognizer exclusively using MFCC features. Column B shows the results of the exemplar-based SC system, where the full 179-dimensional state-likelihoods were used for decoding with a Viterbi back end. It is evident that the SC based decoder outperforms the traditional recognizer from [7] substantially by 50% absolute at SNR 0dB. However, the SC based decoder does not perform as good in the SNR 20dB and clean conditions.

The last three columns in Table 1 show the results using our dual input MFCC/SC architecture and their labels indicate which feature stream the architecture observes (corresponding to the three cases introduced in Section 4.2).

The *MFCC* column ($\alpha = 1$; $\beta = 0$) can be regarded as the baseline performance of a traditional HMM implemented as a DBN. Its results are comparable with those obtained in [10]. The DBN has a more powerful back-end with more model parameters which results in outperforming the traditional HMM recognizer in column A by around 16% absolute on average. At SNR 0dB, however, it still does not perform as good as the exemplar-based SC method which does about 11% better.

In the SC column ($\alpha = 0$; $\beta = 1$) a drop of the word accuracy relative to column B can be observed. A likely explanation is that some information is lost due to the mapping from 179 states to 20 phones. Since some phones may occur in more than one digit, the mapping from 179 states to 20 phonemes reduces the correlation between states and their corresponding words. Moreover, when mapping the 20-dimensional phone probability vector to a scalar denoting the single most likely phone index, some valuable information about the less likely candidates is lost and this hard evidence (a single discrete index) makes the model vulnerable to any single mistake of the SC approach. Nevertheless, the SC column still shows the same SNR-dependency trend as column B and the recognition accuracy at SNR 0dB is still significantly higher than those in columns A and *MFCC*.

When the DBN uses a 50-50% balance between the two

streams (column *MFCC/SC*), a big improvement over the DBN *MFCC* baseline is achieved. Although at SNR 0dB, the SC method in column B still achieves the highest recognition accuracies of all five approaches, the difference in performance between the dual input *MFCC/SC* and SC method in column B has reduced to a non-significant 1.0%. Thus, *MFCC/SC* exceeds *MFCC* by over 10% absolute at SNR 0dB.

When comparing the MFCC-based approaches to the dual-input MFCC/SC architecture we observe that the latter also produces competing results in cleaner conditions. Apparently the dual input architecture succeeds in combining the best of two worlds and compensates for the shortcomings of the SC system underlying columns B and SC. Moreover, the dual input model outperforms either of the single input models *MFCC* and *SC* at all the other SNRs, suggesting that the used information streams are truly complementary.

4.4. Weighting MFCC and SC observations

Since MFCC and SC based architectures perform differently in different conditions, we expected the trustworthiness of either feature to be dependent on SNR. Therefore, we studied in more detail how the performance of the dual-input DBN architecture varies as a function of both SNR and the relative weight of each feature stream. We used the *MFCC/SC* model from Table 1, where both MFCC and SC inputs had a 0.5 weight during training, and repeated the decoding with varying weights. We varied the MFCC weight α from 0 to 1 and the SC weight $\beta = 1 - \alpha$ correspondingly using a step size of 0.1.

Table 2: *Impact of the relative weights of the MFCC and SC feature streams on word recognition accuracy scores (in %). The 0.95 confidence intervals (assuming a binomial distribution) are printed between brackets.*

MFCC	*SC*	SNR0	SNR10	SNR20	clean
0	1	61.2 (1.5)	87.4 (1.0)	92.0 (0.8)	93.5 (0.8)
0.1	0.9	68.0 (1.4)	91.7 (0.9)	96.1 (0.6)	97.5 (0.5)
0.2	0.8	**69.6 (1.4)**	93.0 (0.8)	97.2 (0.5)	98.4 (0.4)
0.3	0.7	69.3 (1.4)	93.5 (0.8)	97.6 (0.5)	98.6 (0.4)
0.4	0.6	68.3 (1.4)	**93.7 (0.8)**	97.7 (0.5)	98.7 (0.4)
0.5	0.5	66.8 (1.5)	93.4 (0.8)	**97.8 (0.5)**	98.8 (0.3)
0.6	0.4	65.1 (1.5)	93.3 (0.8)	97.7 (0.5)	98.8 (0.3)
0.7	0.3	63.4 (1.5)	92.8 (0.8)	97.7 (0.5)	**98.9 (0.3)**
0.8	0.2	61.7 (1.5)	92.3 (0.8)	97.7 (0.5)	98.7 (0.4)
0.9	0.1	60.4 (1.5)	92.0 (0.8)	97.7 (0.5)	98.7 (0.4)
1	0	58.5 (1.5)	91.3 (0.9)	97.6 (0.5)	98.8 (0.3)

Table 2 shows the word recognition accuracy averaged over the four noise types for each pair of weights. The results show that the *MFCC/SC* weight pairs 0.2/0.8, 0.4/0.6, 0.5/0.5 and 0.7/0.3 achieve the highest accuracies best for the SNR 0dB, 10dB, 20dB and clean conditions, respectively. While the differences are often not significant, there is a clear trend that confirms our expectation that we should trust SC information over MFCC's at low SNRs, but MFCC over SC at high SNRs.

Another interesting observation to be made from these results is that when the weight of MFCC feature stream equals 0, meaning we only use the SC stream, the performance is low. However, as soon as the slightest bit of MFCC information is involved (increasing the MFCC weight from 0 to 0.1), a substantial improvement of more than 4% on average is achieved. This illustrates that the SC index stream misses some important information, either due to the extremely simplified mapping or due to the SC algorithm itself, but which is retained in the MFCC coefficients.

5. Conclusions

In this work, we proposed a dual input architecture for noise robust speech recognition. An HMM is implemented as a DBN allowing us to use a combination of a classical MFCC vector and a phoneme prediction (derived from an exemplar-based sparse classification (SC) algorithm) as input features. Although we severely reduced the dimensionality of the sparse classifier by mapping it to the index of the most likely phoneme, this dual input model still proved capable of combining the noise robustness of the SC approach with the high accuracy at high SNRs of MFCCs.

Based on the obtained results, there are several options for future work. First, a switching parent could be added to the DBN architecture, so that for each frame the decoding weights of SC and MFCC can be adjusted on the fly. The value of this switching parent could for instance be controlled by the output of a module that estimates the SNR of the current utterance. Second, an alternative line of research would be to explore to what extent it is feasible to incorporate a complete probability vector as additional SC input (at the phoneme or even state level).

6. Acknowledgements

The research of Yang Sun is supported by the Marie Curie Initial Training Network SCALE. The research of Jort F. Gemmeke is supported by the Dutch-Flemish STEVIN Programme.

7. References

[1] R. P. Lippmann, "Speech recognition by machines and humans," *Speech Commun.*, vol. 22, no. 1, pp. 1–15, 1997.

[2] O. Scharenborg, "Reaching over the gap: A review of efforts to link human and automatic speech recognition research," 2007.

[3] J. Gemmeke, H. Van hamme, B. Cranen, and L. Boves, "Compressive sensing for missing data imputation in noise robust speech recognition," *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, no. 2, pp. 272–287, 2010.

[4] J. Gemmeke, L. ten Bosch, L.Boves, and B. Cranen, "Using sparse representations for exemplar based continuous digit recognition," in *EUSIPCO*, Glasgow, Scotland, 2009.

[5] J. F. Gemmeke and T. Virtanen, "Noise robust exemplar-based connected digit recognition," in *Proceedings of ICASSP 2010*, Dallas, USA, 2010.

[6] J. Bilmes, "Graphical models and automatic speech recognition," University of Washington, Department of Electrical Engineering, Tech. Rep. UWEETR-2001-0005, 2001.

[7] H. G. Hirsch and D. Pearce, "The AURORA experimental framework for the performance evaluation of speech recognition systems under noisy conditions," in *ISCA ITRW ASR2000*, 2000, pp. 29–32.

[8] M. Wöllmer, F. Eyben, B. Schuller, Y. Sun, and T. Moosmayr, "Robust in-car spelling recognition - a tandem blstm-hmm approach," in *Proc. Interspeech 2009, Brighton, UK.* ISCA, 2009, pp. 2507–2510.

[9] J. Bilmes, "The GMTK documentation," 2002.

[10] C.-P. Chen, J. Bilmes, and K. Kirchhoff, "Low-resource noise-robust feature post-processing on aurora 2.0," in *Proceedings of ICSLP 2002*, 2002, pp. 2445–2448.