Multimodal Speaker Detection using Input/Output Dynamic Bayesian Networks

Vladimir Pavlović[1], Ashutosh Garg[2], and James M. Rehg[1]

[1] Compaq Cambridge Research Lab
Cambridge, MA 02142
{vladimir,rehg}@crl.dec.com
[2] ECE Department and Beckman Institute
University of Illinois
Urbana, IL 61801
ashutosh@ifp.uiuc.edu

Abstract. Inferring users' actions and intentions forms an integral part of design and development of any human-computer interface. The presence of noisy and at times ambiguous sensory data makes this problem challenging. We formulate a framework for temporal fusion of multiple sensors using input–output dynamic Bayesian networks (IODBNs). We find that contextual information about the state of the computer interface, used as an input to the DBN, and sensor distributions learned from data are crucial for good detection performance. Nevertheless, classical DBN learning methods can cause such models to fail when the data exhibits complex behavior. To further improve the detection rate we formulate an *error-feedback* learning strategy for DBNs. We apply this framework to the problem of audio/visual speaker detection in an interactive kiosk application using ”off-the-shelf” visual and audio sensors (face, skin, texture, mouth motion, and silence detectors). Detection results obtained in this setup demonstrate numerous benefits of our learning-based framework.

1 Introduction

Human-centered user-interfaces based on vision and speech present challenging sensing problems. Multiple sources of information, including high-level application-specific information, must be combined to infer the user's actions and intentions. Statistical modeling techniques play a critical role in the design and analysis of such systems. Dynamic Bayesian network (DBN) models are an attractive choice, as they combine an intuitive graphical representation with efficient algorithms for inference and learning. Previous work has demonstrated the power of these models in fusing video and audio cues with contextual information and expert knowledge both for speaker detection and other similar applications [3, 5, 4, 2].

Speaker detection is a particularly interesting example of a multi-modal sensing task with application in video conferencing, video indexing and human-computer interaction. Both video and audio sensing provide important information in a multi-person and noisy scenarios. Contextual information or the state of the application is another important component because it often governs the type of interaction. This naturally leads one to consider a special DBN architecture known as input/output DBN (Figure 2(b)). The state of the application forms an input to the system which is, along with the sensory outputs, used to determine the state of the user. We are interested in network models that combine "off-the-shelf" vision and speech sensing with contextual cues.

Estimation of the DBN model parameters is a key step in the design of the detection system. Strengths of the DBN arcs in Figure 1, from context to task variables to sensors, can be automatically learned from data using standard maximum-likelihood (ML) learning schemes, similar to [3]. However, it is often the case that the chosen model structure only approximately represents the data. To circumvent this drawback we introduce a learning algorithm for DBNs that uses *error-feedback* to improve recognition accuracy of the model. In error-feedback DBNs (EFDBNs) strengths of DBN arcs are iteratively adjusted by focusing on data instances incorrectly detected by the previous models.

This paper demonstrates that modeling the contextual states of the application as an input to a DBN together with the learning of continuous sensor distributions can enhance performance of DBNs which fuse temporal data from weak multimodal sensors. We also show how EFDBN learning strategy yields significant improvements in detection accuracy. We present these results in the context of a network architecture of Figure 1 which infers the state of the speaker who actively interacts with the Genie Casino game. Our evaluation of the learned DBN model indicates its superiority over previous static [7] and dynamic [3, 5] detection models.

2 Speaker Detection

An estimate of the persons state (whether s/he is or isn't a speaker) is important for the reliable functioning of any speech-based interface. We argue that for a person to be an active speaker, s/he must be expected to speak, face the computer system and actually speak. Visual cues can be useful in deciding whether the person is facing the system and whether he is moving his lips. However, they are not capable on their own to distinguish an active user from an active listener (listener may be smiling or nodding). Audio cues, on the other hand, can detect the presence of relevant audio in the application. Unfortunately, simple audio cues are not sufficient to discriminate a user in front of the system speaking to the system from the same user speaking to another individual. Finally, contextual information describing the "state of the world" also has bearing on when a user is actively speaking. For instance, in certain contexts the user may not be expected to speak at all. Hence, audio and visual cues as well as the context need to be used jointly to infer the active speaker.

We have analyzed the problem of speaker detection in a specific scenario of the Genie Casino Kiosk. The Smart Kiosk [6] developed at Compaq's Cambridge Research Lab (CRL) provides an interface which allows the user to interact with the system using spoken commands. This version of kiosk simulates a multiplayer blackjack game (see Figure 4 for a screen capture.) The user uses a set of spoken commands to interact with the dealer (kiosk) and play the game. The kiosk has a camera mounted on the top that provides visual feedback. A microphone is used to acquire speech input from the user.

We use a set of five “off-the-shelf” visual and audio sensors: the CMU face detector [8], a Gaussian skin color detector [10], a face texture detector, a mouth motion detector, and an audio silence detector. A detailed description of these detectors can be found in [7]. Contextual input provides the state of the application (the blackjack game) which may help in inferring the state of the user.

2.1 Bayesian networks for speaker detection with continuous sensors and contextual input

We adopt a modular approach towards the design of the Bayesian network for speaker detection. We have designed modules for vision and audio tasks separately which are then integrated along with the higher level information.

The graph in Figure 1 shows the vision network for this task. This network takes the output of the vision sensors and outputs the query variables corresponding to visibility and the frontal information of the user. Face detector gives a binary output whereas the output of the texture detector is modeled as a conditional Gaussian distribution whose parameters are *learned* from the training data. We contrast this to the cases studied in [3, 5] where all sensors had binary outputs.

The audio network combines the output of the silence detector and the mouth motion detector. These detectors provide continuous valued output as the measure of the silence and mouth motion respectively. The audio network selected for this task is shown in Figure 1. The output of the audio network corresponds to the probability that the audio in the application corresponds to the user present.

Once constructed, the audio and visual networks are fused to obtain the integrated audio–visual network. The contextual information acts as an input to the model with the sensory observations as the output. The state of the user (e.g. speaker vs. nonspeaker) forms the state of the model and needs to be inferred given the observations and the inputs. The final network obtained is shown in Figure 1.

The final step in designing the topology of the speaker detection network involves its temporal aspect. Measurement information from several consecutive time steps can be fused to make a better informed decision. This expert knowledge becomes a part of the speaker detection network once the temporal dependency shown in Figure 2(a) is imposed. Incorporating all of the above elements into a single structure lead to the input/output DBN shown in Figure 2(b). The input/output DBN structure is a generalization of the input/output HMM [1]—here the probabilistic dependencies between the variables are governed by the BN shown in Figure 1. The speaker node is the final speaker detection query node.

The use of continuous valued sensor outputs allows the network to automatically learn optimal sensor models and, in turn, optimal decision thresholds. In the previous work [3, 5] sensor outputs were first discretized using decision thresholds set by expert users. Here all continuous sensory outputs are modeled as conditional Gaussian distributions, as shown in Figure 3. The learned distributions allow soft sensory decisions which can be superior to discrete sensory outputs in noisy environments. Indeed, results outlined later in this paper show that improved performance is obtained using this model.

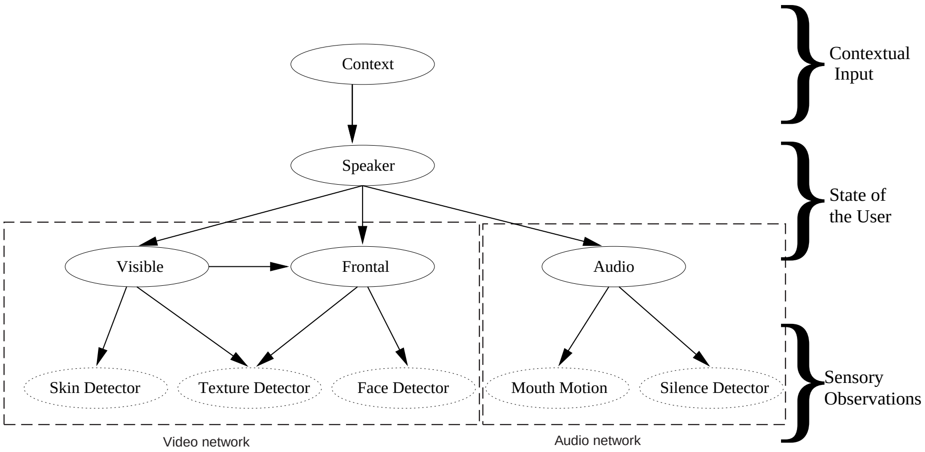


Fig. 1. Integrated audio-visual network.

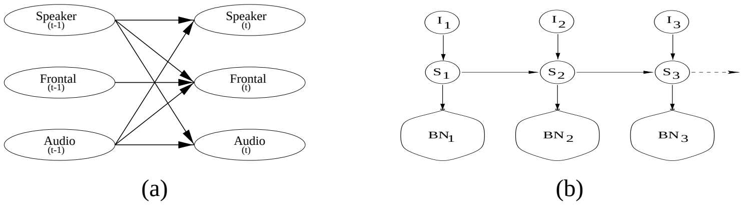
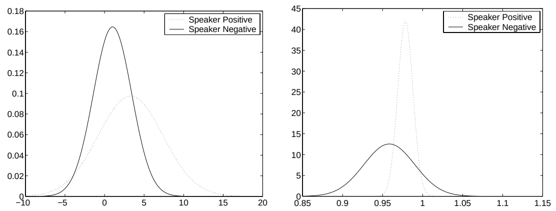


Fig. 2. (a) Temporal dependencies between the speaker, audio, and frontal nodes at two consecutive time instances. (b) The input output DBN, with contextual information as input and the sensory observations as outputs.

3 Learning dynamic Bayesian networks

Dynamic Bayesian networks are a class of Bayesian networks specifically tailored to model temporal consistency present in data. In addition to describing dependencies among different static variables DBNs describe probabilistic dependencies among variables at different time instances. A set of random variables at each time instance t is represented as a static BN. Out of all the variables in this set temporal dependency is imposed on some. Thanks to its constrained topology efficient inference and learning algorithms, such as forward-backward propagation and Baum-Welch, can be employed in DBNs (see [5] for more details.)

However, classical DBN learning algorithms assume that the selected generative model accurately represents the data. This is often not the case as the selected model is only an approximation of the true process. Recently, Schapire et al. [9] have proposed a method called *boosting* aimed at improving the performance of any simple (classification) model. In particular, their *Adaboost* algorithm “boosts” the classification on a set of data points by linearly combining a number of weak models, each of which is trained to correct “mistakes” of the previous one. In a similar spirit we formulated the



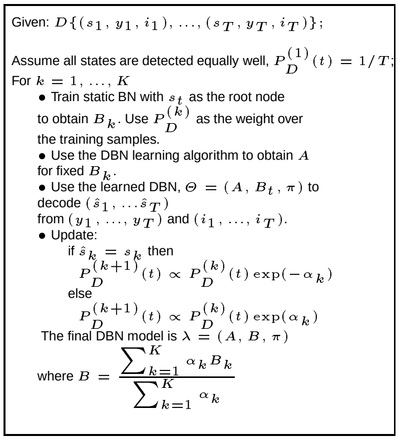
(a) (b) (c)

Fig. 3. Learned continuous sensor distributions: (a) silence, (b) skin texture, and (c) mouth motion.

Error Feedback DBN framework [5]. Here we extend this framework to handle the case of continuous sensory outputs and contextual input.

3.1 Error Feedback DBNs

Consider the training data $D = \{(s_1, y_1, i_1), ..., (s_T, y_T, i_T)\}$ of duration T, where s denotes DBN states, i are the inputs, and y are the measurements, and the DBN shown in Figure 2(b). The goal of DBN learning is to, given data D, obtain the DBN model $\Theta = (A, B, \pi)$, (where A is the transition probability matrix dependent on input i, B is the observation matrix which maps s_t to y_t and π is the initial distribution of s_0) which minimizes the probability of classification error in s on dataset D. EFDBN algorithm for this setting can be formulated as follows.


```
Given: $D\{(s_1, y_1, i_1), \ldots, (s_T, y_T, i_T)\}$;

Assume all states are detected equally well, $P_D^{(1)}(t) = 1/T$;
For $k = 1, \ldots, K$
    • Train static BN with $s_t$ as the root node
    to obtain $B_k$. Use $P_D^{(k)}$ as the weight over
    the training samples.
    • Use the DBN learning algorithm to obtain $A$
    for fixed $B_k$.
    • Use the learned DBN, $\Theta = (A, B_t, \pi)$ to
    decode $(\hat{s}_1, \ldots \hat{s}_T)$
    from $(y_1, \ldots, y_T)$ and $(i_1, \ldots, i_T)$.
    • Update:
        if $\hat{s}_k = s_k$ then
            $P_D^{(k+1)}(t) \propto P_D^{(k)}(t) \exp(-\alpha_k)$
        else
            $P_D^{(k+1)}(t) \propto P_D^{(k)}(t) \exp(\alpha_k)$
    The final DBN model is $\lambda = (A, B, \pi)$
    where $B = \dfrac{\sum_{k=1}^{K} \alpha_k B_k}{\sum_{k=1}^{K} \alpha_k}$
```

The algorithm maintains a weight distribution defined over the data. It starts by assigning equal weight to all the samples. As the algorithm proceeds, the weight of correctly classified samples is decreased whereas that of misclassified ones is increased. Details of the original algorithm can be found in [5].

4 Experiments and Results

We conducted three experiments using a common data set. The data set comprised of five sequences of a user playing the blackjack game in the Genie Casino Kiosk setup. The experimental setup is depicted in Figure 4. The same figure shows some of the recorded frames from the video sequence. Each sequence included audio and video tracks recorded through a camcorder along with frequency encoded contextual information (see Figure 4.) The visual and audio sensors were then applied to audio and video streams. Examples of individual sensor observations (e.g., frontal v.s. non frontal, silence v.s. non silence, etc.) are shown in Figure 5. Abundance of noise and ambiguity in these sensory outputs clearly justifies the need for intelligent yet data-driven sensor fusion.

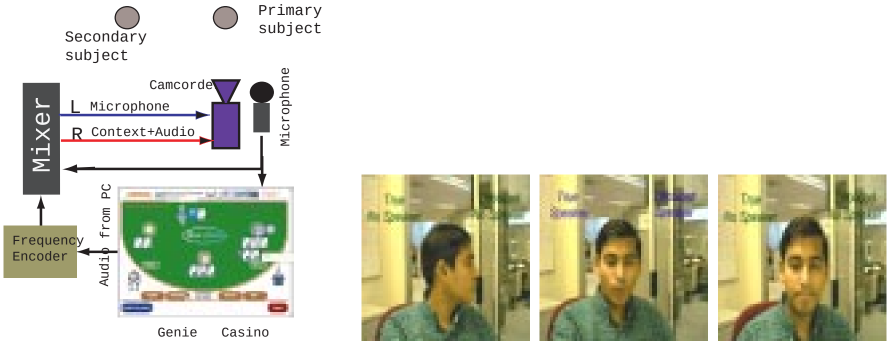
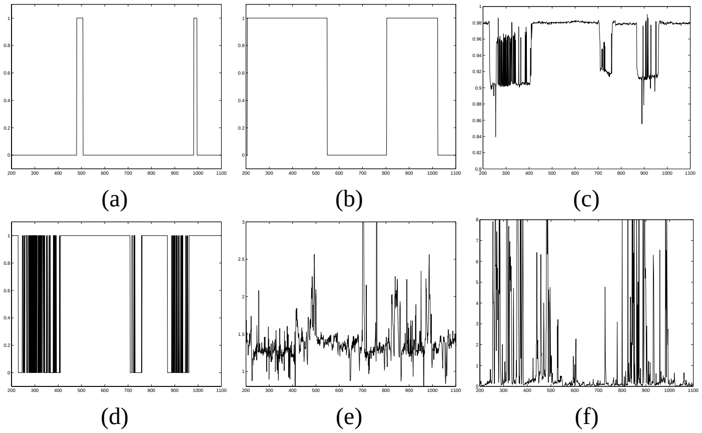


Fig. 4. Data collection setup for Genie Casino kiosk and three frames from a test video sequence.

4.1 Static Bayesian network

The first experiment was done using the static BN of Figure 1 to form the baseline for comparison with the dynamic model. In this experiment all samples of each sequence was considered to be independent of any other sample. Part of the whole data set was considered as the training data and rest was retained for testing. During the training phase, output of the sensors along with the hand labeled values for the hidden nodes (speaker, frontal and audio) were presented to the network.

During testing only the sensor outputs were presented and inference was done to obtain the values for the hidden nodes. Mismatch in any of the three (speaker, frontal, audio) is considered to be an error. An relatively low average accuracy of 77% is obtained (see Figure 6 for results on individual sequences.) The sensor data (as shown in Figure 5) is noisy and it is hard to infer the speaker without making substantial errors. Figure 7(a) shows the ground truth sequence for the state of the speaker and (b) shows the decoded sequence using static BN. However, this detection accuracy is superior to 68% obtained in [3, 5] when discrete sensors and non-input context were used.



(a) (b) (c)

(d) (e) (f)

Fig. 5. (a) Ground truth for the speaker state: 1 indicates the presence of the speaker. (b) Contextual information: 1 indicates user's turn to play. (c),(d),(e),(f) Outputs of texture, face, mouth motion and silence detectors, respectively.

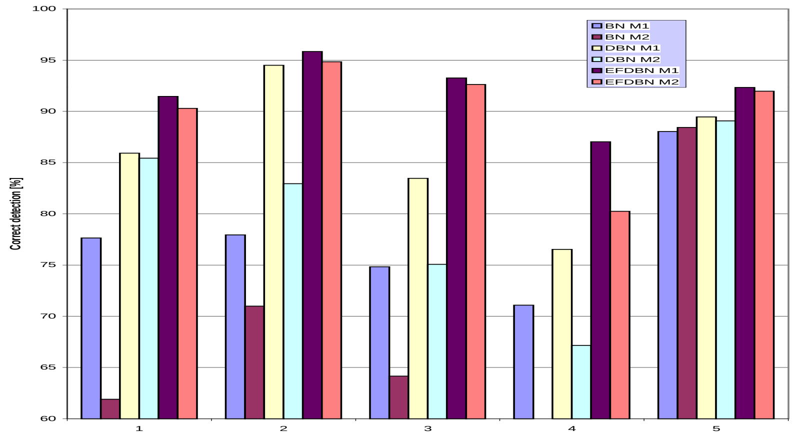
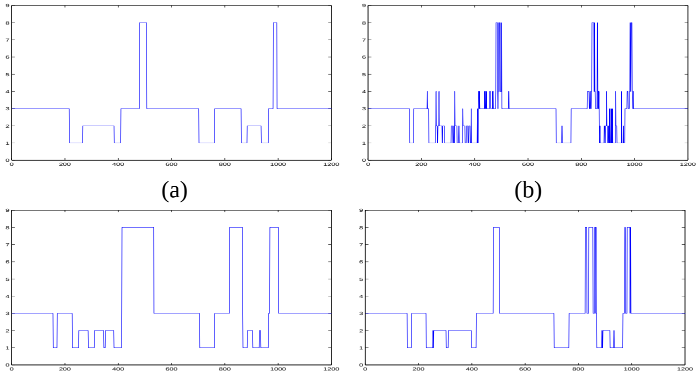


Fig. 6. A comparison between the results obtained using static BN, DBN, and EFDBN. M1 and M2 denote the models with continuous and discrete sensory inputs, respectively.

4.2 Input/output DBN

Second experiment was conducted using the input/output DBN model. The standard ML learning algorithm described in Section 3 was employed to learn the dynamic transitional probabilities among frontal, speaker, and audio states. During testing phase a temporal sequence of sensor values was presented to the model and Viterbi decoding



(a) (b) (c) (d)

Fig. 7. (a) True state sequence. (b),(c),(d) Decoded state sequences by static BN, DBN, EFDBN, respectively. (state 1 - no speaker, no frontal, no audio; state 2 - no speaker, no frontal, audio; state 3 - no speaker, frontal, no audio; state 8 - speaker, frontal, audio)

was used to find the most likely sequence of the speaker states. Overall, we obtained the accuracy of the speaker detection of 85%, an improvement of $\approx 10\%$ over the static BN model. An indicative of this can be seen in actual decoded sequences in Figure 7. The improved performance by the use of DBN stems from the inherent temporal correlation present between the features. Again, the use of continuous sensors and input context produced significant improvement compared to 80% rate of [3, 5].

4.3 Error-feedback DBN

Our final experiment employed the newly designed EFDBN framework for continuous sensors and contextual application input. The learning algorithm described in Section 3.1 was used. For a training sequence, we used EFDBN to estimate the parameters which minimized the classification error. A leave-one-out crossvalidation resulted in the overall accuracy of 92%. Figure 6 summarizes classification results on individual sequences. We see that for all the sequences, an improvement of $5 - 10\%$ over the best DBN result is obtained. While the improvement is less dramatic over the 90% detection rate of the EFDBN with discrete sensors and contextual measurement [5], it still remains significant.

The DBN model learned using the EFDBN framework was also applied to the prediction of hidden states. An overall accuracy of 88% was obtained. This indicates, together with the previously noted results, that EFDBN significantly improves the performance of simple DBN classifiers. In comparison of the results with the ones reported in [3, 5], we observe that significant improvement in performance is obtained. This can be attributed to the use of continuous sensory observations and input/output DBN structure.

5 Discussions and Conclusions

We have presented a general purpose framework for learning input/output DBN models for fusion of continuous sensory output and contextual, application-specific input. The framework encompasses a new error-feedback learning procedure which can circumvent the effects of simple models and complex data. The results obtained for the difficult problem of speaker detection where a number of noisy sensor outputs need to be fused indicate the utility of this algorithm. Significant improvements in classification accuracy over a simple DBN model were achieved without sacrificing of complexity of the learning algorithm. We have also demonstrated a general purpose approach to solving man-machine interaction tasks in which DBNs are used to fuse the outputs of simple audio and visual sensors while exploiting their temporal correlation.

Reliability and confidence of sensors during inference is one crucial aspect of sensor fusion tasks which was not addressed in this framework. For instance, the number of skin colored pixels in the whole image can be used as a measure of the reliability of the skin sensor and hence weigh its contribution relative to other sensors. Future research will focus on incorporating sensor reliabilities into our current framework. Another interesting opportunity in this DBN framework arises as a consequence of modeling application-specific context as input. Namely, one can study methods of designing contextual input which will force the user from its present state (e.g., non-speaking) to a new desired state in a number of steps. These opportunities become even more significant when the number of system states becomes large, a case often encountered in dialog systems.

References

[1] Y. Bengio and P. Frasconi, “An input-output HMM architecture,” in *Advances in Neural Information Processing Systems 7*, pp. 427–434, Cambridge, MA: MIT Press, 1995.

[2] M. Brand, N. Oliver, and A. Pentland, “Coupled hidden markov models for complex action recognition,” in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, (San Juan, PR), pp. 994–999, 1997.

[3] A. Garg, V. Pavlovic, J. Rehg, and T. S. Huang, “Audio–visual speaker detection using dynamic Bayesian networks,” in *Proc. of 4rd Intl Conf. Automatic Face and Gesture Rec.*, (Grenbole, France), pp. 374–471, 2000.

[4] S. Intille and A. Bobick, “Representation and visual recognition of complex, multi-agent actions using belief networks,” Tech. Rep. 454, MIT Media Lab, Cambridge, MA, 1998.

[5] V. Pavlovic, A. Garg, J. Rehg, and T. S. Huang, “Multimodal speaker detection using error feedback dynamic Bayesian networks.” To appear in Computer Vision and Pattern Recognition 2000.

[6] J. M. Rehg, M. Loughlin, and K. Waters, “Vision for a smart kiosk,” in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, (Puerto Rico), pp. 690–696, 1997.

[7] J. M. Rehg, K. P. Murphy, and P. W. Fieguth, “Vision-based speaker detection using bayesian networks,” in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, (Ft. Collins, CO), pp. 110–116, 1999.

[8] H. Rowley, S. Baluja, and T. Kanade, “Neural network-based face detection,” in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, (San Francisco, CA), pp. 203–208, 1996.

[9] R. E. Schapire and Y. Singer, “Improved boosting algorithms using cofidence rated predictions.” To appear in Machine Learning.

[10] J. Yang and A. Waibel, “A real-time face tracker,” in *Proc. of 3rd Workshop on Appl. of Comp. Vision*, (Sarasota, FL), pp. 142–147, 1996.