Audio–Visual Speaker Detection using Dynamic Bayesian Networks

Submission no. 182

Abstract

The development of human-computer interfaces poses a challenging problem: actions and intentions of different users have to be inferred from sequences of noisy and ambiguous sensory data. Temporal fusion of multiple sensors can be efficiently formulated using dynamic Bayesian networks (DBNs). DBN framework allows the power of statistical inference and learning to be combined with contextual knowledge of the problem. We demonstrate the use of DBNs in tackling the problem of audio/visual speaker detection. "Off-the-shelf" visual and audio sensors (face, skin, texture, mouth motion, and silence detectors) are optimally fused along with contextual information in a DBN architecture that infers instances when an individual is speaking. Results obtained in the setup of an actual human-machine interaction system (Genie Casino Kiosk) demonstrate superiority of our approach over that of static, context-free fusion architecture.

1. Introduction

Advanced human–computer interfaces increasingly rely on sources of multiple yet often unreliable information. Ambiguity and noise embedded in such sources make the use of statistical inference crucial for interface applications. We address the application of dynamic Bayesian network (DBN)models [3] to the task of detecting whether a user is speaking to the computer.

DBNs are a class graphical probabilistic models, derived from the better known Bayesian networks (c.f. [8, 7]). Bayesian networks have been successfully employed in a wide range of expert system and decision support applications. One example is the Lumière project [6] at Microsoft, which used Bayesian networks to model user goals in Windows applications. DBNs graphically encode dependencies among sets of random variables who evolve in time. They elegantly combine the benefits of both data- and expert-driven models. On one hand, the structure of dependencies among variables can be a priori determined by an expert designer who has the knowledge of the task domain. On the other, the strength of those influences can be learned from large sets of data. Some applications of DBNs can be found in [1].

In this paper we demonstrate the use of DBNs in fusing multiple visual and audio sensors, contextual information, temporal constraints and one's expert knowledge in solving the challenging speaker detection problem. We improve the static network architecture of [12] through a network, shown in Figure 6, which dynamically combines the outputs of five simple "off-the-shelf" algorithms to detect the presence of a speaker. The structure of the network encodes the context of the sensing task and knowledge about the operation of the sensors. The conditional probabilities along the arcs of the network relate the sensor outputs to the task variables. These probabilities are learned automatically from training data. We have analyzed this problem in a an interactive scenario of a Genie Casino Kiosk [11, 2] which plays a multi-agent blackjack game with a human user.

This paper makes a contribution by demonstrating the representational strength of DBNs in fusing temporal data coming from different weak sensors with the expert knowledge of the task domain and contextual state of the environment. We present a network architecture of Figure 6 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 BN models [12].

2. Speaker Detection Problem

Speaker detection is a fundamental problem in any human-centered computer system. We argue that for a person to be an active user (speaker), he must be expected to speak, facing the system and actually speaking. 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 environment. 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.

The Smart Kiosk [11, 2] developed at Compaq's Cambridge Research Lab (CRL) provides an interface which allows the user to interact with the system using spoken commands. The public, multi-user nature of the kiosk application domain makes it ideal as an experimental setup for speaker detection task. The kiosk (see Figure 1(a)) has a camera mounted on the top that provides visual feedback. A microphone is used to acquire speech input from the user. This setup forms an ideal testbed for our problem.

We have analyzed the problem of speaker detection in a specific scenario of the Genie Casino Kiosk. This version of kiosk simulates a multiplayer blackjack game (see
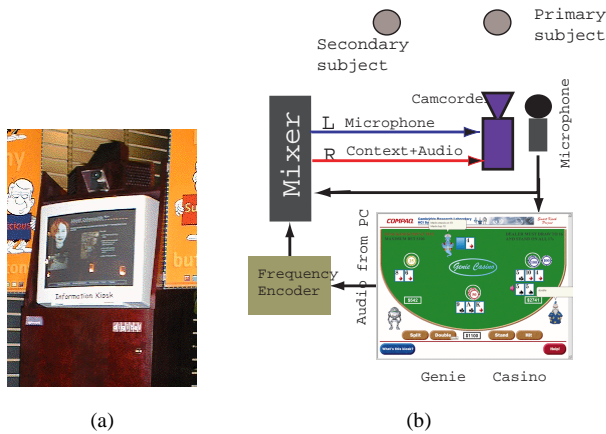
Figure 1. The Smart Kiosk (a) and Experimental setup for data collection (b).

Figure 1(b).) The user uses a set of spoken commands to interact with the dealer (kiosk) and play the game.

2.1. Sensors

Audio and visual information can be obtained directly from the two kiosk sensors. We use a set of five "off-the-shelf" visual and audio sensors: the CMU face detector [13], a Gaussian skin color detector [15], a face texture detector, a mouth motion detector, and an audio silence detector. These components have the advantage of either being easy to implement or easy to obtain, but they have not been explicitly tuned to the problem of speaker detection. A more detailed description of skin, texture, face and mouth motion detectors can be found in [12].

Contextual sensor is, as it will become clear later, of utmost importance. It provides the state of the environment which may help in inferring the state of the user. Contextual information can tell whether the user is expected to speak or not. For example, when a computer has asked the user for some information, the likelihood of the user speaking increases. On the contrary, when the computer is answering some simple query made by the user the likelihood of an active speaker decreases. In our setup we select a simplified state of the game as the contextual information. Namely, two game states are encoded: the user's turn (to interact) and its complement. Onsets of contextual states are marked with beeps of specific frequencies. To avoid possible distraction the frequency encoded contextual signal is directly sent to the camcorder without being played on the speakers (see Figure 1(b).)

3. Dynamic Bayesian Networks for Speaker Detection

Dynamic Bayesian networks are a class of Bayesian Networks specifically tailored to model temporal consistency present in some data sets. Bayesian networks (BN) (c.f. [8, 7]) are a convenient *graphical way* of describing dependencies among random variables. Variables are represented as nodes in directed acyclic graphs whose arc "weights" correspond to conditional probability distributions (tables or functions) among dependent variables. See [8, 7] for a thorough coverage of this subject.

There are two computational tasks that must be performed in order to use BNs as classifiers. After the network topology (dependency among variables) has been specified, the first task is to obtain the local conditional probability tables (CPTs) for each variable conditioned on its parent(s). Once the CPTs have been specified (either through learning or from expert knowledge), the remaining task is inference, i.e., computing the probability of one set of nodes (the query nodes) given another set of nodes (the evidence nodes). In our speaker detection the evidence nodes are the discretized outputs of the sensors and the query node is the probability of a detected speaker. See [7] for more details on the standard BN algorithms.

In addition to describing dependencies among different static variables DBNs [3] describe probabilistic dependencies among variables at different time instances. In general, a DBN has a specific structure shown in an example in Figure 6. 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. Namely, distribution of some variable $x_{i,t}$ at time t depends on a variable at time $t-1$, $x_{j,t-1}$ through some conditional distribution $\Pr(x_{i,t}|x_{j,t-1})$. An example of this structure is depicted in Figure 6. Probability distribution among all variables in a DBN can in general be written as $\Pr(x_1, \ldots, x_T) = \Pr(x_1) \prod_{t=2}^{T} \Pr(x_t|x_{t-1})$. It is worth noting that some specific stochastic time-series models now classified as DBNs have been known for many years. For instance, linear dynamic systems and hidden Markov models [9] are indeed special cases of DBNs with continuous and discrete variables, respectively.

A major benefit of the DBNs is that their well-constrained network structure allows for simplified inference. Sometimes complex inference in general BNs reduces to a two-step generalized *forward-backward* message passing procedure in DBNs [4]. An example of this technique is well known from HMM literature. In linear dynamic systems, for instance, these procedures are better known under their Kalman filtering and smoothing names. While the inference in DBNs may reduce to these simple techniques, the BN origin of the model still allows for a plethora of other BN-only inference techniques to be adapted to DBNs. In particular, in DBN structures more complex than an LDS or HMM other inference techniques may be necessary that do not exist in the simple models. More details can be found in [5, 1].

Besides the more constrained structure, another crucial assumption that justifies special treatment of DBNs lies in the fact that entries (parameters) of conditional distributions associated with the network are (almost always) assumed not to vary over time, e.g., $\Pr(x_{i,t}|x_{j,t-1})$ depends only on i, j and not on $t, t-1$. This allows for a very compact representation of DBNs. Together with the simplified inference, compact representation allows for efficient EM learning algorithms to be applied. Sufficient statistics required by EM learning need only be computed within one and across two consecutive time slices. An example of this is Baum-Welch learning algorithm in HMMs [9]. As in the case of inference, learning in complex DBN structures can benefit from

their BN-origins [5, 1].

Our speaker detection problem represents a challenging ground for testing the representational power of DBN models in a complex multi-sensor fusion task. Different types of sensors need to be seamlessly integrated in model that both reflects the expert knowledge of the domain and the sensors and benefits from the abundance of observed data. We approach the model building task by first tackling the expert design of networks that fuse individual sensor groups (video and audio). We then proceed with the integration of these sensor networks with each other, with contextual information, and over time. Finally, data-driven aspect comes into play with data-driven parameter learning.

3.1. Visual Network

Vision network models the dependence between the various observations made by the vision sensors. We want to use the various vision sensors to infer when the user is facing the kiosk and when he is near but not facing it directly. To accomplish that a small BN which takes the binary output of these sensors as its input and outputs the query variables corresponding to visibility and the frontal information of the user is designed. This network structure is also known as a polytree and is depicted in the graph shown in Figure 2. The "visible" and "frontal" are not directly observed but are instead inferred from sensory data. Our expert knowledge leads us to the topology of the network. The user being "frontal" clearly depends on whether he is "visible". If the user is "visible" parts of his skin and face will appear in the image. On the other hand, the face detection sensor only detects frontal faces. Hence, it is plausible to connect it to the "frontal" node. Probability distribution defined
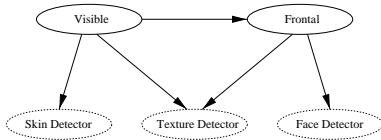

Figure 2. Vision Network

by the visual network is now $\Pr (V, F, SK, TX, FD) = \Pr (V)\Pr (F|V)\Pr (SK|V)\Pr (TX|V,F)\Pr (FD|F)$, where V, F, SK, TX, FD correspond to the "visual", "frontal", "skin", "texture", and "face detector" nodes, respectively. If one were simply to use the visual network as the speaker detector, the posterior distribution of interest would be $\Pr (F|SK, TX, VD)$. This posterior can be efficiently obtained using a number of BN inference techniques (e.g., junction tree potentials). The optimal Bayesian decision that the the speaker was present is then made if $\Pr (F = \text{true}|SK, TX, VD) > Pr(F = \text{false}|SK, TX, VD)$.

3.2. Audio Network

The other components of sensory data available provide what we call the audio information. Namely, the silence detector and the mouth motion detector are used to infer whether the user is talking. Silence detector is used to detect audio signals present in the environment. To discriminate between the audio signal who originates in some back-

ground source (which may be noise) and the one coming as a result of the user speaking, we use a vision sensor (mouth motion detector) to supplement the audio signal. The resulting audio network is shown in Figure 3. Binary "audio" query node captures the information about the user talking. This network demonstrates the fusion of the audio and the visual information at a very low level. Although the mouth motion detector is a vision sensor, it is more closely related to the silence detector than to other vision sensors. Probability distribution defined by the audio network is sim-
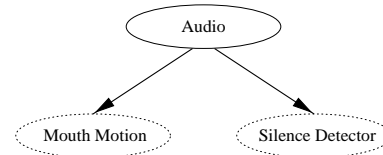

Figure 3. Audio network for speaker detection.

ply $\Pr (A, MM, SL) = \Pr (A)\Pr (SL|A)\Pr (MM|A)$, where A, MM, SL denote "audio", "mouth motion"', and "silence" nodes. If this network alone were to be used as the speaker detector, optimal decision could be made by comparing $\Pr (A = \text{true}|MM, SL)$ to $\Pr (A = \text{false}|MM, SL)$.

3.3. Integrated Audio–Visual Network

Once constructed, the audio and visual networks are fused to obtain the integrated audio–visual network. At this stage one would also like to incorporate any information the environment may play in deciding the user's state. The contextual information (state of the blackjack game), together with the visual and audio subnetworks is now fused into a single net through the virtue of the speaker node, as shown in Figure 4. The chosen network topology represents our knowledge that both audio, visual, as well as contextual conditions need to be met for the decision on the presence of the speaker to be made: along the course of the game when the user is expected to speak he should be facing the kiosk and talking. To infer whether a user is speaking, one
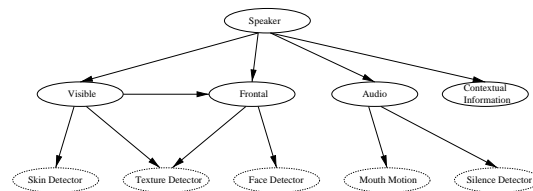

Figure 4. Integrated audio-visual network.

needs to find the posterior $\Pr (S|SK, TX, FD, MM, CT)$. Again, this posterior can be easily obtained using a number of BN inference techniques.

3.4. Dynamic Network

The final step in designing the topology of the spekear detection network involves its temporal aspect. The character of the observation processes involved justifies the use of temporal dependencies. As it will be shown in the experimental section, noise and ambiguity of individual sensors

at different time instances may lead to incorrect inference about the speaker. Fortunately, decisions about the speaker (as well as the frontal and audio states) does not change rapidly over time. In fact, if it is know with certain probability that the speaker is present *before* and *after* some time t a better informed decision can be expected to be made about the speaker at the time t. Equivalently, 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 5 is imposed. The
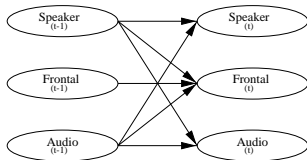

Figure 5. Temporal dependencies between the speaker, audio, and frontal nodes at two consecutive time instances.

presence of all possible arcs among the three nodes stems from our lack of exact knowledge about these temporal dependencies, i.e., we allow for all dependencies to be present and later on determined by the data. Formal techniques for this network *structure learning* can be found in [1].

Incorporating all of the elements above elements into a single structure lead to the DBN shown in Figure 6. Here the nodes shown in dotted lines are the direct observation nodes while the ones in solid are the unobserved nodes. The speaker node is the final speaker detection query node. Inference in this network now corresponds to finding the
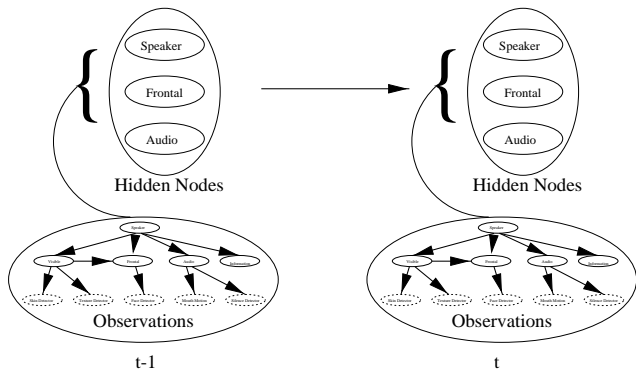

Figure 6. Two time slices of the dynamic Bayesian network for speaker detection.

distribution of the speaker variable S_t at each time instance conditioned on a sequence of measurements from the sensors, $\mathcal{M}_1^T = \{SK_1, TX_1, FD_1, MM_1, CT_1, \ldots, SK_T, TX_T, FD_T, MM_T, CT_T\}$. Optimal detection of the speaker at time t can now be made by comparing $\Pr(S_t = \text{true}|\mathcal{M}_1^T)$ to $\Pr(S_t = \text{false}|\mathcal{M}_1^T)$. These posteriors are obtained directly from the forward-backward inference algorithm. One may also be interested in predicting the likelihood of the speaker from all the previous observation, $\Pr(S_{t+1} = \text{true}|\mathcal{M}_1^t)$.

3.5. Learning

Given the topology of the DBN discussed in the previous sections learning of network parameters can formulated as the maximum likelihood parameter problem. A straightforward application of the EM algorithm for DBNs can iteratively lead to (locally) optimal CPTs that agree with the data.

To further simplify the learning procedure we isolated the learning of the "observation" portions of the DBN from the dynamic, transition CPTs. Namely, we first learned the "observation" network CPTs assuming no temporal dependencies, and then employ the fixed "observation" networks to learn the transition probabilities. While obviously suboptimal this procedure has been shown in practice to yield good parameter estimates that do not differ significantly from the optimal ones.

At this stage we draw the attention to a fact that in DBNs, the probability of staying in a certain state over consecutive time instances decays exponentially. In the speaker detection problem, this may not be a prefered model. Namely, in the constrained environment of a blackjack game durations of certain states tend to be fairly well defined. For instance, duration of words in the small vocabulary of the dealer agent effectively defines the duration of contextual states. These states, in turn, have significant bearing on when a user is the speaker. Thus, we also explored *duration density* DBN (DDDBN) models where state (speaker, frontal, audio) durations were explicitly modeled. Inference in these models becomes untractable as the length of the duration model increases. We adopted the techniques suggested in [10] for doing learning and inferencing in the DDDBN.

4. Experiments and Results

We conducted three experiments were conducted using a common data set. The data set comprised of five sequences of one user playing the blackjack game in the Genie Casino setup. The sequences were of varying duration (from 2000 samples to 3000 samples) totaling to 12500 frames. Figure 7 shows some of the recorded frames from the video sequence. Each sequence included audio and

Figure 7. Three frames from a test video sequence.

video tracks recorder through a camcorder along with frequency encoded contextual information (see Figure 1(b).) The visual and audio sensors were then applied to audio and video streams. Because some of the sensors provide continuous estimates of their respective functions (e.g., silence sensor's internal output is the short-term energy of the audio signal), decision thresholds were determined for each sensor that yield binary sensor states (e.g., silence v.s. no

silence.) These discretized states were the used as input for the DBN model. Examples of individual sensor decisions (e.g., frontal v.s. non frontal, silence v.s. non silence, etc.) are shown in Figure 8. Abundance of noise and ambiguity in these sensory outputs clearly justifies the need for intelligent yet data-driven sensor fusion.
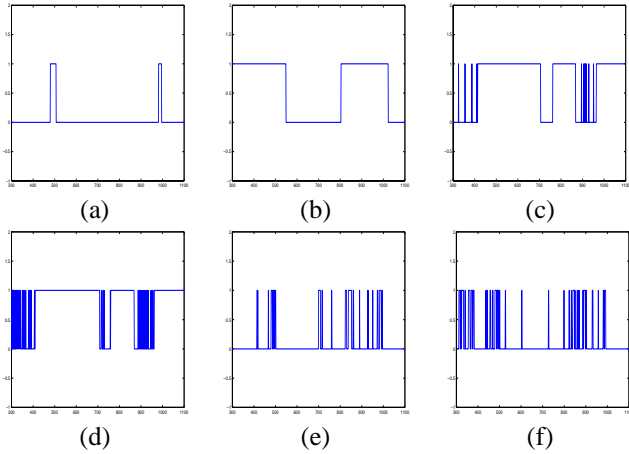

Figure 8. Figure (a) shows the ground truth for the speaker state. 1 means that there is a speaker and 0 means an absence. x axis gives the frame no. in the sequence. (b) gives the contextual information. 1 means, its users turn to play where as 0 means the computer is going to play. (c),(d),(e),(f) are the output of texture, face, mouth motion and silence detector respectively.

4.1. Experiment Using Static BN

The first experiment was done using the static BN Figure 4 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 label values for the hidden nodes (speaker, frontal and audio) node were presented to the network. The network does learn CPTs that are to be expected. The actual CPT values show that the presence of the speaker (S=1) must be expressed through the presence of a talking (A=1) frontal face (F=1) in the appropriate context of the game (C=1). On the other hand, the existence of the frontal face alone does not necessarily mean that the speaker is present (S=0,F=1).

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. Cross validation was done by choosing different and training and test data. An average accuracy of 65% is obtained (see Figure 9 for results on individual sequences.) The accuracy is low even though the learned network parameter do seem intuitive, as explained above. Figure 8 depicts a typical output of the sensors along with the ground truth for the speaker state. The sensor data is noisy and it is hard to infer the speaker without making substantial errors. Figure 10(a) shows the ground truth
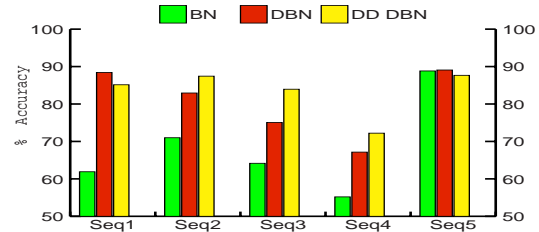

Figure 9. A comparison between the results obtained using static BN, DBN, DDDBN

sequence for the state of the speaker and (b) shows the decoded sequence using static BN. On the other hand, temporal consistency in the query state (speaker ground truth) indicates that a model should be built that exploits this fact.
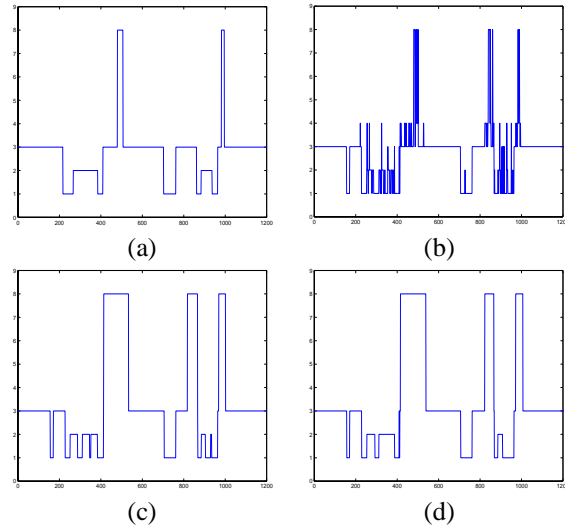

Figure 10. Figure (a) shows the true state sequence. (b),(c),(d) are the decoded state sequences by static BN, DBN, DDDBN respectively. (state 1 - no speaker, no frontal, no audio; state 2 - no speaker, no frontal, audio; state 3 - no speaker, frontal, no audio; state 8 - speaker, frontal, audio)

4.2. Experiment Using DBN

Second experiment was conducted using the DBN model. At sequence level data was considered independent (e.g. seq1 is independent of seq2.) The learning algorithm described in Section 3.5 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 (c.f. [9]) was used to find the most likely sequence of the speaker states. Overall, we obtained the accuracy of the speaker detection (after cross validation) of about 80%, an improvement of 15% over the static BN model. An indicative of this can be seen in actual decoded sequences. For

instance, decoded sequence using the DBN model in Figure 10 is obviously closer to the ground truth than the one decoded using the static model.

It is clear why the DBN model performed better than the static one. Inherent temporal correlation of features was indeed exploited by the DBN. This can be confirmed by expecting the learned CPT of temporal transitions among S, A, and F nodes.

4.3. Experiment Using DDDBN

We finally tested the representational power of the DDDBN approach. Duration densities for state durations of one up to twenty were learned from the labeled data. Figure 11 shows the learned CPTs. The four states for these graphs are plotted are: (a) no speaker, no frontal, no audio, (b) no speaker, no frontal, audio, (c) no speaker, frontal, no audio, and (d) speaker, frontal, audio. It is evident from these graphs that some of the duration distributions clearly differ from exponential distribution imposed by the DBN model. Our speaker detection accuracy indeed gets improved when this model is used. An average accuracy of 83% is obtained. Figure 10 (d) shows an example of the decoded state sequence using the DDDBN model. Nonetheless, improved performance of the DDDBN model
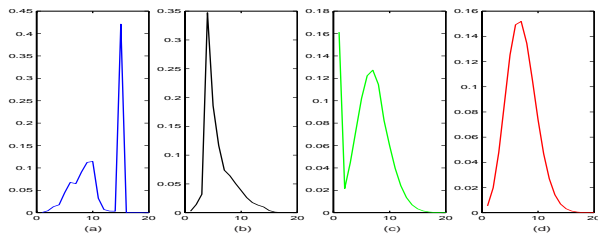

Figure 11. Duration density plots for states of the DDDBN model. Shown are: (a) no speaker, no frontal, no audio, (b) no speaker, no frontal, audio, (c) no speaker, frontal, no audio, and (d) speaker, frontal, audio states.

is severely hampered by its complexity. The complexity of inference in DDDBNs increases exponentially with the duration of the states (compared to a DBN). In practice, this prevents one from using DDDBN models and favors the simpler yet almost as powerful DBNs.

5. Conclusions

We have 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. DBNs provide an intuitive graphical framework for expressing expert domain knowledge and temporal consistency of processes coupled with efficient algorithms for learning and inference. They can represent complex models of stochastic processes, but their learning rules are simple closed-form expressions given a fully-labeled data set.

Simpler static multisensor fusion models based on BNs have been introduced before (e.g. [12]). By using DBNs to impose models of temporal consistency already present in the task we have shown that significant improvements in performance can be made over that of the static models. Our speaker detection experiments using the network of Figure 6 demonstrated classification rates of 85%. The advantage of the principled and well-defined DBN framework will become even more obvious as the complexity of tasks scales upward. Data- and expert-driven DBNs will provide a viable alternative to often encountered complex and ad-hoc algorithms whose design is exclusively determined by the knowledge of an expert user.

In future work we will further validate our network designs on a large subject population under realistic conditions of background clutter. We will also investigate improvements on our sensor models. Finally, we plan to step beyond a single decision maker and engage the power of a pool of expert models [14] to better infer complex variable dependencies.

References

[1] X. Boyen, N. Firedman, and D. Koller, "Discovering the hidden structure of complex dynamic systems," in *Proc. Uncertainty in Artificial Intelligence*, pp. 91–100, 1999.

[2] A. D. Christian and B. L. Avery, "Digital smart kiosk project," in *ACMSGICHI*, (Los Angeles, CA), 1998.

[3] T. Dean and K. Kanazawa, "A model for reasoning about persistance and causation," *Computational Intelligence*, vol. 5, no. 3, 1989.

[4] B. Frey, *Graphical Models for Machine Learning and Digital Communication*. MIT Press, 1998.

[5] Z. Ghahramani, "Learning dynamic Bayesian networks," in *Adaptive processing of temporal information* (C. L. Giles and M. Gori, eds.), Lecture notes in artificial intelligence, Springer-Verlag, 1997.

[6] E. Horvitz, J. Breese, D. Heckerman, D. Hovel, and K. Rommelse, "The Lumière project: Bayesian user modeling for inferring the goals and needs of software users," in *Proc. of the 14th Conf. on Uncertainty in AI*, pp. 256–265, 1998.

[7] F. V. Jensen, *An introduction to Bayesian Networks*. Springer-Verlag, 1995.

[8] J. Pearl, *Probabilistic reasoning in intelligent systems*. San Mateo, CA: Morgan Kaufmann, 1998.

[9] L. R. Rabiner and B. Juang, *Fundamentals of Speech Recognition*. Englewood Cliffs, New Jersey, USA: Prentice Hall, 1993.

[10] P. Ramesh and J. G. Wilpon, "Modeling state durations in hidden Markov models for automatic speech recognition," in *Proc. IEEE Int'l Conference on Acoustics, Speech, and Signal Processing*, 1992.

[11] J. M. Rehg, M. Loughlin, and K. Waters, "Vision for a smart kiosk," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, (St. Juan, PR), 1997.

[12] J. M. Rehg, K. P. Murphy, and P. W. Feiguth, "Vision-based speaker detection using bayesian networks," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, (Ft. Collins, CO), 1999.

[13] H. Rowley, S. Baluja, and T. Kanade, "Neural network-based face detection," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 203–208, 1996.

[14] R. E. Schapire, "A biref introduction to boosting," in *Proc. Int'l Joint Conference on Artificial Intelligence*, (Stockholm, Sweden), 1999.

[15] 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.