DYNAMIC BAYESIAN NETWORKS FOR MEETING STRUCTURING

Alfred Dielmann and Steve Renals

Centre for Speech Technology Research
University of Edinburgh
Edinburgh EH8 9LW, UK
Email: {a.dielmann,s.renals}@ed.ac.uk

ABSTRACT

This paper is about the automatic structuring of multiparty meetings using audio information. We have used a corpus of 53 meetings, recorded using a microphone array and lapel microphones for each participant. The task was to segment meetings into a sequence of meeting actions, or phases. We have adopted a statistical approach using dynamic Bayesian networks (DBNs). Two DBN architectures were investigated: a two-level hidden Markov model (HMM) in which the acoustic observations were concatenated; and a multistream DBN in which two separate observation sequences were modelled. Additionally we have also explored the use of counter variables to constrain the number of action transitions. Experimental results indicate that the DBN architectures are an improvement over a simple baseline HMM, with the multistream DBN with counter constraints producing an action error rate of 6%.

1. INTRODUCTION

Meetings are sociological events, in which a large amount of information is generated and shared between a group of participants. They are so important that, for example, software houses specialized in large software projects allocate half their budget to organize meetings. Therefore an automated system to capture, store, structure and index meetings, could be useful to:

- spread knowledge between people who have missed the meeting
- preserve meeting contents, avoiding confusions and omissions, enabling meeting participants to recall details
- understand the structure of meetings (temporal evolution, decision taking processes, etc.)

Initially we can simply capture meeting contents, through multi-perspective and multi-channel audio-video recordings. However, without further analysis, the semantic content of the meeting remains locked in an intractable low-level multimodal data stream. Text transcriptions of speech in meetings (eg, the ICSI meeting project [1]) are a further step in this task. Meetings are a case of spontaneous human interaction, and their transcriptions tend to be redundant and only partially able to highlight meeting structure.

In this work we use a dictionary of group "meeting actions" (such as monologues, dialogue between participants, note taking, presentations and presentations at a white-board) which may be considered both as group social actions and as meetings phases [2]. These meeting actions may be used to segment meetings, identifying different communicative phases. The meeting action sequence provides a description of meeting structure, and builds up a simple semantic language, which may be used to formulate queries for a retrieval system, or to assist meeting browsing.

This paper is organized as follows. In section 2 we present an overview of the recognition of group meeting actions, the set of chosen features and the adopted meeting corpus. In section 3 we describe two Dynamic Bayesian Network models developed by us. Finally in section 4 we compare experimental results, achieved using proposed models to segment the corpus in term of meeting actions.

Supported by EU IST project M4 (IST–2001–34485). We thank Iain Mc Cowan of IDIAP for providing the speech activity features.

2. MEETING ACTION RECOGNITION

Meeting actions may be performed by individual participants or jointly by a group of participants. Such actions may be characterized across several modalities, such as speech, gesture, facial expression and body movement. The whole communication process is distributed across several modalities, hence a multimodal approach is required in order to obtain a comprehensive view of meetings. In this work we have used audio data, in particular location information and prosodic features to segment meetings into a sequence of five possible meeting actions: monologue, dialogue, note taking, presentation, presentation at the white-board [2].

2.1. Experimental data

We used the publicly available meeting corpus[1] collected by IDIAP as part of the (IM)2 and M4 projects, using an instrumented "Smart Meeting Room"[3]. This corpus consists of 53 short meetings[2] (average length: 5 minutes) with 4 participants per meeting. The whole corpus consists of about 2 hours of multi-channel audio/visual recordings. The video was captured using three fixed cameras, audio was recorded using one lapel microphone for each participant, and an eight element circular microphone array. Meeting structure was generated a priori, using a set of seven meeting actions: monologue (four possible speakers), dialogue, note taking, presentation, presentation at the white-board, consensus and disagreement. Each meeting was composed of an average of five actions, with dialogue being the most frequent action. Therefore the meeting structure was scripted, but participants' behaviors were natural

[1] http://mmm.idiap.ch

[2] The work reported in [2] used a set of 60 meetings; however 7 of those meetings are no longer available

and recording conditions realistic. We subdivided the available set of meetings in two parts: the first 30 meetings were used training, and the remaining 23 meetings formed the test set.

2.2. Features

Since speech is the predominant communicative modality in meetings, we concentrated on audio features. In particular we have used speaker turn features and prosodic features. Speaker turns are useful to highlight which participant has the focus of attention, and how the conversation evolves in time. For example, looking for patterns in speaker turns, we could attempt to discriminate between monologues and dialogues.

Speaker turn features were extracted using the microphone array: the spatial diversity of microphones may be exploited through the use of a beam-forming process. It is possible to estimate sound source directions, assigning a probability to each direction. These measures are then integrated, considering only those regions of space where meeting participants are expected to be located [4]; during these meetings people spend most of the time in their seat, presenting, or writing on the white-board. Assuming that every participant occupies one seat and there are 2 presentation spaces, we have 6 different location based "speech activities":

$$L_i(t) \;\; \forall i \in [1,6]$$

A 216–dimension speaker turn feature vector was constructed, formed from the 6^3 possible products of 6 location based speech activities, in a temporal window of 3 frames :

$$S_{ijk}(t) = L_i(t) \cdot L_j(t-1) \cdot L_k(t-2) \;\; \forall i,j,k \in [1,6]$$

The prosodic features were extracted from lapel microphone signals. Three feature types were considered: root mean square signal energy, baseline pitch and rate of speech. F0 was estimated using the ESPS pitch extraction algorithm[3] then filtered with an histogram filter, a median filter and an interpolating filter [5]. This filtering chain stylizes and denoises the F0 contour, removing local ripples and unwanted peaks. Syllable rate was estimated through the use of Multiple RATE estimator [6]. All three acoustic features were computed for each participant, resulting in an acoustic feature vector of 12 elements. These features are masked with the speech activity calculated through beam-forming. Therefore features are greater than zero only if the corresponding speaker is active. Both speaker turns and acoustic features were down-sampled in order to share the same sampling frequency of 2Hz.

3. DYNAMIC BAYESIAN NETWORK MODELS

A Bayesian Network (BN) is a convenient graphical way to describe statistical dependencies between a set of variables: variables are indicated with nodes and directed edges represent the influence that each variable has on the others. If there is no edge between two nodes, then the corresponding variables are conditionally independent given the other variables. A Dynamic Bayesian Network (DBN) generalizes a BN by representing how a set of random variables may evolve over time. A static BN is instantiated for each temporal slice t and oriented arcs connect variables of different time-slices (BNs). Hidden Markov Models (HMMs), coupled HMMs, factorial HMMs and many other statistical models are particular cases of DBNs [7]. The use of a such unified graphical/mathematical formalism presents many advantages: it highlights the internal model structure, makes it easier to construct a common view of different models and makes it easier to develop new models. Furthermore the Graphical Model ToolKit (GMTK) [8] is a publicly available software package to perform inference and decoding of such models[4].

As a baseline model for our experiments, we chose an ergodic HMM. If we consider its graphical representation, this model contains only two nodes: the hidden state A which represents "meeting actions", and the observable feature vector Y. In this case speaker turns and prosodic features are merged together in a 228-dimension feature vector ("early integration"). This basic model is not only the baseline for our experiments, but also the starting point for other two models that we used.

3.1. Two-level HMM

The two-level HMM (figure 1(a)) is designed to decompose meeting actions as sequences of sub-actions, factorizing the state space and representing it with two levels of resolution (actions A and sub-actions S). Each action is responsible for a set of sub-actions, and these are mapped into observable feature vectors Y. As can be seen in the lower part of figure 1 (comprising the relations between A, S and Y), there is a hierarchy of two coupled ergodic HMM chains, as in Hidden Markov Decision Trees [9]. The lower level models dependences between continuous features Y_t and hidden discrete sub-actions S_t, through the use of a Gaussian mixture model (GMM). The top level maps sub-actions S_t into meeting actions A_t, and is modelled using a conditional probability table (CPT). The joint distribution for a sequence of T temporal slices is:

$$P(A_{1:T}, S_{1:T}, Y_{1:T}) = P(A_1) \cdot P(S_1 \mid A_1) \cdot P(Y_1 \mid S_1) \cdot \\ \cdot \prod_{t=2}^{T} P(A_t \mid A_{t-1}) \cdot P(S_t \mid A_t, S_{t-1}) \cdot P(Y_t \mid S_t) \quad (1)$$

The probability action states at the start time $P(A_1)$ is obtained by training like the other conditional probabilities: $P(A_t \mid A_{t-1})$, $P(S_1 \mid A_1)$, $P(S_t \mid A_t, S_{t-1})$ and $P(Y_t \mid S_t)$. The model appears to be similar to a 2-level hierarchical HMM (HHMM), but in HHMM there is an additional node that enable state changes in the top chain only when the lower chain has reached an "exit state", and a *vertical transition* is possible [10]. Here there is nothing forcing the top chain to change more slowly than the bottom one. Note also that different actions are free to share the same sub-action, and that the number of available sub-actions is a model parameter. Sub-actions are thus obtained as a result of the training process, and do not necessarily have not an obvious interpretation.

3.2. Counter structure

Figure 1 (b) depicts an additional *counter* structure that has been appended to the two-level HMM, utilizing counter variables C and enabler variables E. This counter structure was appended to the model in order to develop a model of the expected number of recognized actions. In this case we may regard the action variables A as generating a sequence of hidden counter variables, in addition to the sequence of observations. The influence of the action variables on the counter variables is mediated by the enabler variables

[3] Available from http://www.speech.kth.se/software

[4] http://ssli.ee.washington.edu/∼bilmes/gmtk/

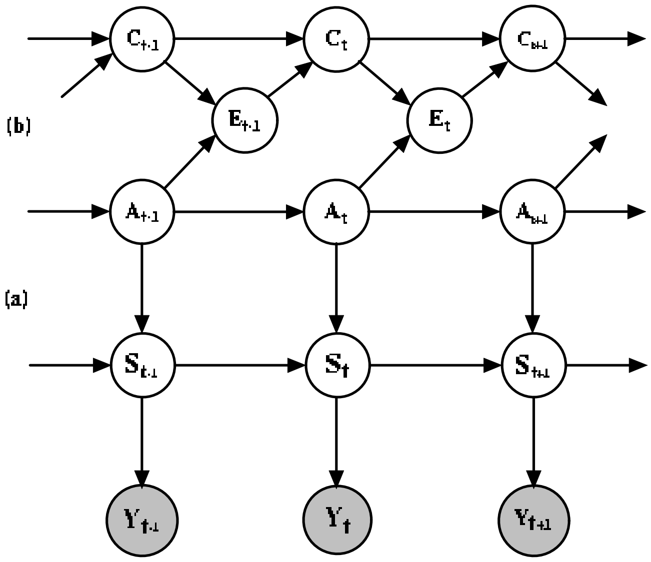

Fig. 1. Two-level HMM (a) with counter structure (b).

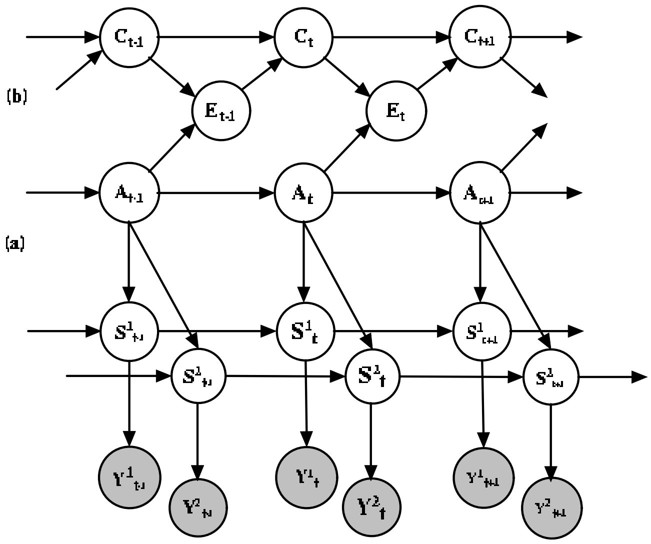

Fig. 2. Multistream DBN model (a) with counter structure (b)

E. The value of E_t depends on both the previous value of E_{t-1} and the last value of C_{t-1}. C_t is a hidden discrete variable that counts the number of recognized actions, and is therefore incremented only if $E_t = 1$. During the training phase given the list of actions, E is imposed to 1 when a transition from one meeting action to another one occurs, and C is progressively increased. Behaviors of E_t and C_t learned during the training phase, are then exploited during the decoding, making the model time-variant. Therefore after the integration with the counter structure, the equation (1) must be multiplied by:

$$P(C_1) \cdot P(E_1) \cdot \prod_{t=2}^{T} P(C_t \mid C_{t-1}, E_{t-1}) \cdot P(E_t \mid C_t, S_t) \qquad (2)$$

obtaining the conditional probability $P(E_t \mid C_t, S_t)$ as a result of the training process, and assuming $P(C_1 = 0) = 1$, $P(E_1 = 0) = 1$, $P(C_t = i+1 \mid C_{t-1} = i, E_{t-1} = 1) = 1$

3.3. Multi stream DBN model

A limitation of the previously presented model is that both speaker turns and prosodic features are integrated into a single feature vector prior to modelling. An alternative approach is to process independently features of a different nature, and integrate them at a higher level of the model. This approach is employed in a second *multistream* DBN model (figure 2(a)). Each feature group Y^1 and Y^2 is processed independently and modeled with hidden sub-actions states S^1 and S^2. Therefore we have two (or eventually more) independent HMM chains and each one is responsible only for a part of the feature set. The top chain, represented by the hidden action node A_t, is responsible to model the whole meeting action, inferring it from the state of two sub-actions nodes. Hence node A could be seen as the integration point. Given a sequence of T frames, the joint distribution is given by:

$$\begin{aligned} P(A_{1:T}, S^1_{1:T}, S^2_{1:T}, Y^1_{1:T}, Y^2_{1:T}) &= P(A_1) \cdot P(S^1_1 \mid A_1) \cdot \\ &\cdot P(S^2_1 \mid A_1) \cdot P(Y^1_1 \mid S^1_1) \cdot P(Y^2_1 \mid S^2_1) \cdot \\ &\cdot \prod_{t=2}^{T} \{ P(A_t \mid A_{t-1}) \cdot P(S^1_t \mid A_t, S^1_{t-1}) \cdot P(S^2_t \mid A_t, S^2_{t-1}) \cdot \\ &\cdot P(Y^1_t \mid S^1_t) \cdot P(Y^2_t \mid S^2_t) \} \,. \qquad (3) \end{aligned}$$

Conditional probabilities are represented through CPTs and learned, as usual, during the training. As in the previous model, the cardinalities of S^1 and S^2 are model parameters. As was done before for the two-level HMM, it is possible to append a counter structure (figure 2(b)). The new joint distribution in this case may be obtained by multiplying together equation (3) and equation (2).

4. EXPERIMENTAL RESULTS

Our evaluations were conducted on the released part of IDIAP meeting corpus, all models were trained and tested using GMTK toolkit [8]. After some initial experiments, we decided to recognize only 5 of the meeting actions used to transcribe the corpus: audio derived features alone are insufficient to model "agreement" and "disagreement" events [2]. The symbols that we would like to recognize are high level symbols and therefore transcription boundaries are only approximate. Being interested in the recognition of the correct actions sequence rather than on the precise time alignment of recognized action segments, the metric that we adopted to evaluate system performances is the Action Error Rate:

$$AER = \frac{Substitutions + Deletions + Insertions}{Correct\, number\, of\, actions}$$

This metric is equivalent to the Word Error Rate used in speech recognition, and is more severe than the frame-based classification accuracy. Table 1 shows experimental results achieved using: a baseline HMM, the two-level model, the multi-stream model and their counter variants.

	Corr	Sub	Del	Ins	AER
HMM	62.1	12.1	25.8	14.4	52.3
two-level	93.2	2.3	4.5	4.5	11.4
two-level + counter	89.4	5.3	5.3	0.8	11.4
multi-stream	90.9	2.3	6.8	2.3	11.4
multi-str. + counter	94.7	1.5	3.8	0.8	6.1

Table 1. Action error rates (%) for the five models trained on the IDIAP meeting corpus training set (30 meetings), and tested on a 23 meeting test set.

	M	DI	NT	PR	WH	INS
M	36		1			2
DI	1	35				2
NT			1			1
PR				12		
WH				1	16	
DEL	1	3	1	1		

Table 2. Confusion matrix of recognized meeting actions for the two-level model, showing monologues (M), dialogues (DI), note taking (NT), presentations (PR), presentations at the white-board (WH), insertion errors (INS) and deletion errors (DEL). Columns show desired symbols and rows obtained actions. Empty cells represent zero values.

The baseline HMM has the lowest recognition accuracy and a very high number of insertions and deletions. Both the two-level model and the multi-stream model have comparable performances: a similar AER, with similar recognition accuracies, but a different balance between insertions and deletions. Adding a counter structure reduces the number of insertions for both models. In the case of the two-level HMM, the reduction in insertions when the counter structure is used is counter balanced by an increase in substitutions and deletions, leaving the AER unchanged. In the case of the multistream model, the AER is significantly reduced in addition to the insertion rate.

To further analyze the results, we give the confusion matrices for the two-level model (table 2) and for the multi-stream model integrated with the counter structure (table 3). The reduction in the number of errors is clearly evident, if we compare the matrices. It is also clear that the note taking action is the least frequent action (only 1.18% of the available corpus), and the most confused symbol, and even the multi-stream model does not recognize it at all. Monologues and presentations at the white-board are the better represented actions, and also the ability to discriminate between monologues and dialogues is excellent (especially in the multi-stream model).

5. CONCLUSIONS AND FUTURE DIRECTIONS

We have presented a method in which meetings are structured as a sequence of actions or phases. We have used audio information only: speaker turn features using location-based speaker activity detection extracted from a microphone array; and, a set of prosodic features (pitch, energy and rate of speech). We have developed and implemented two DBN models for meeting action recognition, tested and trained on the IDIAP meetings corpus. Our results have indicated that the DBN approach to this problem is simple and effective, with action error rates of 6–11%. In the future, we plan to work on larger, more realistic meetings corpora (with an extended set of meeting actions), use more extensive audio-derived features (eg speech recognition transcripts, durational information), and use features derived from video. The multistream DBN developed in this work provides a good platform for these extensions.

	M	DI	NT	PR	WH	INS
M	38		1			1
DI		36				
NT						
PR				12		
WH				1	16	
DEL		2	2	1		

Table 3. Confusion matrix of recognized meeting actions for the multi-stream model integrated with the counter structure.

6. REFERENCES

[1] N. Morgan, D. Baron, S. Bhagat, H. Carvey, R. Dhillon, J. Edwards, D. Gelbart, A. Janin, A. Krupski, B. Peskin, T. Pfau, E. Shriberg, A. Stolcke, and C. Wooters, "Meetings about meetings: research at ICSI on speech in multiparty conversations," *Proc. IEEE ICASSP*, 2003.

[2] I. McCowan, S. Bengio, D. Gatica-Perez, and G. Lathoud, "Modelling human interaction in meetings," *Proc. IEEE ICASSP*, 2003.

[3] D. C. Moore, "IDIAP smart meeting room," *IDIAP COM 02-07*, 2002.

[4] I. McCowan, D. Gatica-Perez, S. Bengio, and G. Lathoud, "Automatic analysis of multimodal group actions in meetings," *IDIAP RR 03-27*, May 2003, Submitted to IEEE Transactions of Pattern Analysis and Machine Intelligence.

[5] K. Sonmez, E. Shriberg, L. Heck, and M. Weintraub, "Modelling dynamic prosodic variation for speaker verification," *Proc. ICSLP*, vol. 7, no. 920, pp. 3189–3192, 1998.

[6] N. Morgan and E. Fosler-Lussier, "Combining multiple estimators of speaking rate," *Proc. IEEE ICASSP*, pp. 729–732, 1998.

[7] P. Smyth, D. Heckerman, and M. I. Jordan, "Probabilistic independence networks for hidden Markov probability models," *Neural Computation*, vol. 9, no. 2, pp. 227–269, 1997.

[8] J. Bilmes and G. Zweig, "The graphical model toolkit: an open source software system for speech and time-series processing," *Proc. IEEE ICASSP*, Jun. 2002.

[9] M. I. Jordan, Z. Ghahramani, and L. K. Saul, "Hidden Markov decision trees," *Proc. of Advances in Neural Information Processing System*, vol. 9, 1996.

[10] S. Fine, Y. Singer, and N. Tishby, "The hierarchical Hidden Markov Model: Analysis and applications," *Machine Learning*, vol. 32, no. 1, pp. 41–62, 1998.