An Adaptive Machine Director

Timothy M. Hospedales
Queen Mary University of London
tmh@dcs.qmul.ac.uk,

Oliver Williams
Microsoft research
olliew@microsoft.com

Abstract

We model the class of problem faced by a video broadcast director, who must act as an active perception agent to select a view of interest to a human from a range of possibilities. Real-time learning of a broadcast direction policy is achieved by efficient online Bayesian learning of the model's parameters based on intermittent user feedback. In contrast to existing machine direction systems, which are dedicated to a particular scenario, our novel approach allows flexible learning of direction policies for novel domains or for viewer-specific preferences. We illustrate the flexibility of our approach by applying our model to a selection of scenarios with audio-visual input including teleconferencing, meetings and dance entertainment.

1 Introduction

In live video broadcast (e.g., on television), the job of a broadcast director is to provide views of interest to a human audience from a range of possibilities. A director will instruct a cameraman to steer his camera to frame salient parts of a scene and, when there are multiple cameras, he must also choose which is the best available view for broadcast. When doing this, it is important that the view is changed in a pleasing way (e.g., without steering or switching too rapidly) and textbooks, such as [2], provide good videography policies for human directors to follow in common scenarios. Attempts have been made to engineer expert machine directors to automate direction in various specific settings. We briefly describe three scenarios which have received interest in automatic broadcasting: lectures, meetings and sporting events.

Recent interest in remote working and learning has made facilitation of live or on-demand Internet broadcast of lectures important. [7] describes a system which directs the broadcast of lectures. This system uses two pan-tilt-zoom (PTZ) cameras: the first camera can be steered to show a room overview or to track the speaker's face; the second can be steered to show questioners. The system implements a direction policy designed by interviewing professional human directors. This policy specifies how the cameras should be individually steered and jointly cut-between given input features such as face detections and microphone array responses. Remote working has also created interest in broadcast and summarization of meetings. [1] describes a machine director which switches between views of each participant and various overview shots. The direction policy is in this case based on participant visibility and speech and motion activity. Automatic camera management for sports broadcasting is also topical. For example, in [3], the authors engineer a digital pan-zoom system to select a salient standard definition window for broadcast given a fixed position high definition video of a soccer event. This system tracks players

BMVC 2008 doi:10.5244/C.22.118

and the ball and uses their locations to compute how to pan and zoom. In each of these cases, significant engineering effort has gone into building a system that is optimized for performing good videography in a specific set of circumstances.

In this paper we describe a probabilistic framework which represents the general class of problems faced by a broadcast director. The parameters of our model are learned from viewer feedback, replacing the need to interview experts and engineer a system for each specific scenario. This means that direction policies can be learned for new or unusual subject domains, for which expert human directors may not exist. Moreover, by rapidly learning the model online, the broadcast policy is customized to fit the viewing preferences of an individual user. We describe particular parametric forms suitable for efficient learning in continuous (camera pan-zoom) and discrete (multi camera switch) domains. Finally, to illustrate the generality and benefits of our learning approach, we apply our adaptive machine director (AMD) to a selection of scenarios including teleconferencing, meetings and dance.

2 Generic Framework

In this section we formalize the task of a director. At each time t the (human or machine) director must decide on the next camera action $\mathbf{d}_t$ to take in response to the currently observed state of the world $\mathbf{s}_t$, and possibly also some of the past history $H_t = \{\mathbf{s}_{1:t-1}, \mathbf{d}_{1:t-1}\}$. The history is necessary to ensure smoothness in the broadcast even when the input state $\mathbf{s}$ is not varying smoothly, and to make some complex direction judgments which require accounting for long range correlations (e.g., avoiding view boredom). We define the *direction policy* π to be the function $\mathbf{d}_t = \pi(\mathbf{s}_t, H_t; \theta)$, parametrized by θ, which specifies the action to take at each time. Videography textbooks [2] specify rule-based forms for π and θ in well-known domains. Like the machine direction systems described in Sec. 1, we use the response of various (potentially salient) feature detectors (such as face, motion and speech detectors) as the input to the model, $\mathbf{s}_t$. The features used are simple and cheap enough for real-time computation and importantly, interpretable, so that for known scenarios, prior knowledge about θ can easily be exploited.

Our goal will be to perform online learning of the parameters θ required for good videography using a small number of user-labeled direction instructions $\{\mathbf{s}_t, \mathbf{d}_t, H_t\}$. To model the changing uncertainty about the policy as more data is observed, we will maintain a distribution over the parameters $p(\theta|\mathbf{d}_{1:t}, \mathbf{s}_{1:t})$. In the case of discrete actions $\mathbf{d}_t$ (such as switching between cameras) the problem of learning θ is related to online semi-supervised classification, while in the case of continuous actions (such as panning and zooming a camera) this problem is related to that of online semi-supervised regression. Fig. 1a illustrates a probabilistic graphical model to represent the broadcast direction problem: in our experiments we model only first-order history $H_t = \{\mathbf{d}_{t-1}\}$. The general procedure for using the adaptive machine director consists of three phases: *prediction*, *validation* and *learning* which are described next.

Prediction At time t, the next action $\mathbf{d}_t$ is selected, based on the observed state $\mathbf{s}_t$ and current policy estimate $p(\theta|\mathbf{s}_{1:t-1}, \mathbf{d}_{1:t-1})$. The posterior over actions is computed as

$$p(\mathbf{d}_t|\mathbf{d}_{1:t-1}, \mathbf{s}_{1:t}) = \int p(\mathbf{d}_t|\theta, \mathbf{s}_t, \mathbf{d}_{t-1}) p(\theta|\mathbf{d}_{1:t-1}, \mathbf{s}_{1:t-1}) d\theta, \quad (1)$$

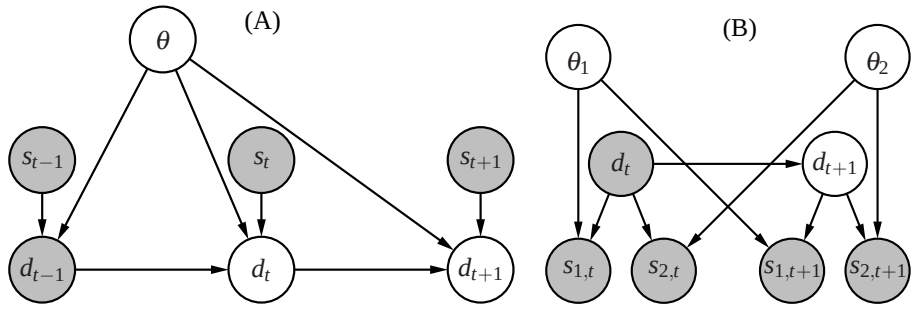

Figure 1: (a) Graphical model to describe the broadcast director with adaptive policy. The direction decisions d_t are and policy parameters θ are to computed online given observations s_t. (b) Alternative graphical model used to describe the broadcast director problem in the discrete case.

from which next action $\mathbf{d}_t$ is sampled and the cameras steered or switched appropriately. Standard, non-adaptive, machine direction systems[7, 3, 1] do not update the parameters θ and effectively implement a deterministic version of this prediction step; we also have validation and learning phases which allow the system to adapt online to user feedback.

Validation As a result of the prediction phase, the broadcast viewed by the user is updated. If she has a strong preference about what she wishes to view she may take manual control to steer or switch the cameras. This is done via a user interface which enables correction of the machine director's decision with simple mouse input. Her instruction thereby labels the desired action $\mathbf{d}_t$. If, however, she is content with the director's decision, she may sit back and continue to watch. In this case, she has also labeled the current action, if only by implicit consent. A key property of the AMD problem scenario is that implicit consent as well as explicit instruction are informative, but the former requires less user effort. We can therefore treat what appears to be a semi-supervised problem as a fully supervised one, albeit with a potentially asymmetrical noise model (see Secs. 3, 4).

Learning Following validation, the correct action $\mathbf{d}_t$, is known either by explicit instruction or implicit consent. The posterior over the direction policy $p(\theta|\mathbf{d}_{1:t}, \mathbf{s}_{1:t})$ can then be updated as

$$p(\theta|\mathbf{d}_{1:t}, \mathbf{s}_{1:t}) \quad = \quad \frac{p(\mathbf{d}_t|\mathbf{d}_{1:t-1}, \mathbf{s}_{1:t}, \theta)p(\theta|\mathbf{d}_{1:t-1}, \mathbf{s}_{1:t-1})}{p(\mathbf{d}_t|\mathbf{d}_{1:t-1}, \mathbf{s}_{1:t})}. \tag{2}$$

There are two key features that make this procedure effective. Firstly, with appropriate parametric choices for the model, the prediction and learning steps can be computationally inexpensive. Secondly, with well chosen features, the user intervention step need only happen very rarely and the system will converge to a well-refined, customized policy with very little cost in terms of both computation and user intervention.

3 Continuous Model for Pan and Zoom Policy

In this section we describe a particular parametric model and some example scenarios where this framework can be used to learn a policy for pan-zoom (PZ) control of a camera. In this case, the decision variable $\mathbf{d}_t = \{d_x, d_y, d_z\}_t$ is real-valued and the director implements a regressor. We define the world state in terms of an M-dimensional vector of real-valued features $\mathbf{s}_t$ representing quantities such as the location and scale of detected motion and human faces etc. Generalized linear models provide a convenient and tractable model for Bayesian regression, and we therefore define the prediction model for each decision dimension i as

$$p(d_{i,t}|\mathbf{s_t}, \theta_i, \varepsilon_i) \quad = \quad \mathcal{N}(d_{i,t}|\theta_i^T \Phi(\mathbf{s}_t), \varepsilon_i^{-1}), \tag{3}$$

where the precision parameter ε_i describes the noise on the decision variables. The basis functions Φ can be any fixed functions of the input state. For many scenarios, the goal is effectively to frame up a salient region of the scene. In these cases the pan-zoom decision is itself a linear function of the state features and linear basis functions $\Phi(\mathbf{x}) = \{1, x_1, .., x_M\}$ can be used. Non-linear basis functions (e.g., Gaussian radial basis functions), permit more complicated non-linear policies to to be learned (see Sec. 3.2) but yield only subtle viewing experience improvement while requiring more data to train. In this paper, we therefore use linear basis functions.

The distribution over the policy parameters (in this case the regression vectors θ_i) is taken to be Gaussian, $p(\theta_i) = \mathcal{N}(\theta_i|\mu_{\theta_i}, \Sigma_{\theta_i})$. The predictive distribution required to perform direction is therefore given by the standard equations for Bayesian linear regression,

$$p(d_{i,t}|d_{i,1:t-1}\mathbf{s}_{1:t}, \varepsilon_i) \quad = \int p(d_{i,t}|\mathbf{s_t}, \theta_i, \varepsilon_i) p(\theta_i|d_{i,1:t-1}, \mathbf{s}_{1:t-1}) d\theta_i = \mathcal{N}(d_{i,t}|\mu_{d_i}, \sigma_{d_i}^2), \tag{4}$$

where $\mu_{d_i} = \mu_{\theta_i}^T \Phi(\mathbf{s}_t)$ and $\sigma_{d_i}^2 = \varepsilon_i^{-1} + \Phi(\mathbf{s}_t)^T \Sigma_{\theta_i} \Phi(\mathbf{s}_t)$ (in which $\mathbf{d}_{t-1}$ is included in $\mathbf{s}_t$ to lighten the notation) . At each time t, the posterior distribution over the policy (weight) vector θ_i is updated in response to the state $\mathbf{s}_t$ and the associated action selected $d_{i,t}$,

$$\begin{aligned} p(\theta_i|d_{i,1:t}, \mathbf{s}_{i,1:t}, \varepsilon_i) \quad &\propto \quad p(d_{i,t}|\mathbf{s}_{i,1:t}, \theta_{i,t}, \varepsilon) p(\theta_i|d_{i,1:t-1}, \mathbf{s}_{i,1:t-1}, \varepsilon_i) = \mathcal{N}(\theta_i|\mu_{\theta_i,t}, \Sigma_{\theta_i,t}) \\ \mu_{\theta_i,t} \quad &= \quad \Sigma_{\theta,t}(\Sigma_{\theta_i,t-1}^{-1} \mu_{\theta_i,t-1} + \varepsilon_i \Phi^T(\mathbf{s}_t) d_{i,t}) \\ \Sigma_{\theta_i,t}^{-1} \quad &= \quad \Sigma_{\theta_i,t-1}^{-1} + \varepsilon_i \Phi(\mathbf{s}_t)^T \Phi(\mathbf{s}_t). \end{aligned} \tag{5}$$

By learning the distribution over weight vectors θ_i the model leans the salient input features (or combination thereof) for a given scenario or user. At the expense of further computation time, the observation noise parameters ε_i could be dealt with automatically in various ways including generalized maximum likelihood. However, for our purposes, it is sufficient to set them empirically. Intervention and implicit consent may not be equally informative (i.e., the user may not always bother to supervise the system when it does an acceptable but non-optimal job). We therefore introduce two parameters ε and ε' for learning in response to active and passively supervised inputs respectively, where $\varepsilon > \varepsilon'$.

3.1 Results

In the following two examples the input features provided are the response from face[5] and motion detectors[6]. These return the position and scale of the region bounding detected faces and motion respectively (or null if nothing is detected). $\mathbf{s}_t$ is therefore a six-dimensional vector when both faces and motion are detected, a three dimensional vector when only one of these is detected and null when neither were detected. A fixed decision tree was used to switch amongst these possibilities and four separate regression models were learned to cover each of these possibilities.

Teleconferencing Scenario In a teleconferencing scenario, it might be desirable to have a PTZ camera (or fixed wide-angle camera), where the camera can mechanically (or digitally) track the user to broadcast a well-framed video of their face. This is trivial to engineer given efficient face detection technology[5]. However, we will illustrate efficient *learning* to track faces, without prior knowledge of their relative saliency and with good videography. Fig. 2 illustrates learning this task starting with an uninformative weight prior. By halfway through the one minute sequence (Fig. 2c) and with only 27 interventions, the model is doing a good job of broadcast direction (Fig. 2a) and needs no more human supervision. It has learned the relevance of faces, and the irrelevance of motion for this task (Fig. 2b). This task is very similar to that of the presenter tracking component in lecture broadcast systems[7], indicating that we could easily learn such scenarios as well.

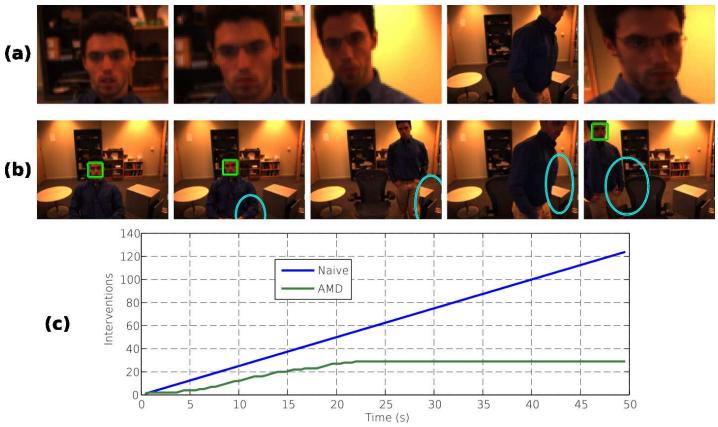

Figure 2: Teleconferencing Scenario. (a) Sample broadcast images. (b) Original images with detected features. (c) Training performance: AMD vs naive full explicit supervision.

Dance Scenario We have mentioned previous applications of machine broadcast directors to entertainment scenarios, specifically soccer[3]. In this scenario we illustrate learning in a novel entertainment scenario, namely dance, using the same two features as before are used. Fig. 3 illustrates learning of this novel scenario starting with an uninformative weight prior. Within 45 seconds and 18 interventions (Fig. 3c), the model is doing a good job of broadcast direction (Fig. 3a). In this case the director learns a policy based largely on framing the region above and around the motion (which is concentrated at the

figures' legs) and ignores (learns zero weight) for the more unreliable face cue. For this second scenario, the cue saliencies are reversed compared to video conferencing.

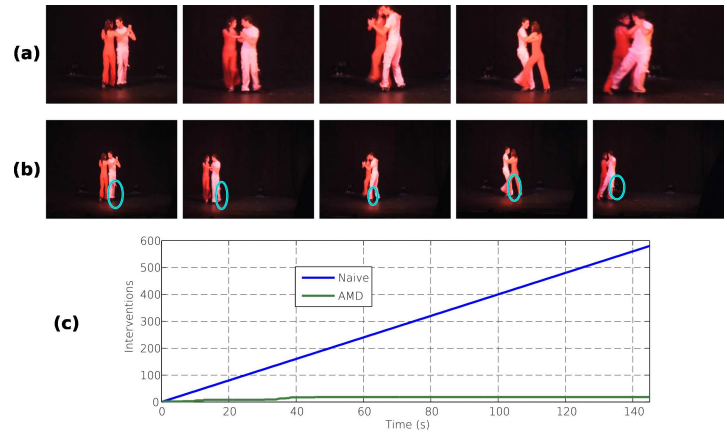

Figure 3: Dancing Scenario. (a) Sample broadcast images. (b) Original images with detected features. (c) Training performance: AMD vs naive full explicit supervision.

3.2 Summary

In this section, we described a class of machine direction problems and solutions where the learning of a direction policy corresponds to learning a regression model. We illustrated that the policy for a teleconferencing problem can be learned online with only a small number of explicit instructions. The simple linear relationship we used between feature values and camera position allows learning standard rules of good direction such as framing up the salient area at a fixed scale, zooming out slightly during periods of high target motion, and zooming out to maximum if the salient target is lost (e.g., Fig. 2, fourth sample frame). Subtler non-linear policies, such as avoiding continuous small corrections by panning to track only when the salient region has left an internal bounding box, can also be learned (using e.g., Gaussian basis functions) but require more training data. The problems considered so far assume a single camera needing to be steered. In the next section, we introduce a discrete valued action model to account for camera switching.

4 Discrete Model for Switching Policy

In this section we describe a structure and particular parametric model with which our framework can be used to learn a policy for switching between multiple cameras using a discrete decision variable $d_t = \{1, .., N\}$ ranging over the number of available cameras N. In this case the M observations in $\mathbf{s}_t$ are discrete variables representing quantities such as the presence or absence of faces, motion or speech activity within the view of each camera. In contrast to Sec. 3's regressor, the direction policy must now implement a dynamic classifier. To minimize the number of policy parameters and maintain an interpretable structure, we will assume independence between the features and consider a committee of simpler models. Although the broadcast direction problem is discriminative, to

avoid the difficulty of rapidly training a product of expert type model online, we move to a generative though *a-causal* model (Fig. 1b). The structure now encodes a Markov model on decisions, where at each time-step observations are generated in a factored (naive-Bayes) way $p(\mathbf{s}_t|d_t) = \prod_i p(s_{i,t}|d_t)$, and we now infer a distribution over actions $p(d_t|\mathbf{s}_t,\theta) \propto p(d_t|\theta)\prod_i p(s_{i,t}|d_t,\theta)$. Conceptually the problem is still the same as Sec. 3 and Fig. 1a; θ still parametrizes a direction policy $d_t = \pi(\mathbf{s}_t, H_t; \theta)$, but now indirectly via the likelihood $p(\mathbf{s}_t|d_t,\theta)$ used during inference of the decision. Eqs. 6 and 7 define a specific multinomial observation and conjugate Dirichlet policy prior for the observation and transition models respectively:

$$\begin{aligned} p(\mathbf{s}_{i,t}|d_t,\theta_{d_t,i}) &= \text{Multi}(\mathbf{s}_{i,t};\theta_{d_t,i}), & p(d_t|d_{t-1},\vartheta_{d_{t-1}}) &= \text{Multi}(d_t;\vartheta_{d_{t-1}}), \\ p(\theta_{d_t,i}) &= \text{Dir}(\theta_{d_t,i};\alpha_{d_t,i}), \quad (6) & p(\vartheta_{d_{t-1}}) &= \text{Dir}(\vartheta_{d_{t-1}};\alpha_{d_{t-1}}). \quad (7) \end{aligned}$$

The parameters $\theta = \{\theta_{d,i}, \vartheta_d\}$ include $\theta_{d,i}$, governing the distribution $p(s_{i,t}|d_t,\theta_{d,i})$ over observations $\mathbf{s}_{i,t}$ in each modality i, and $\vartheta_{d_{t-1}}$, governing the transition between decisions $p(d_t|d_{t-1},\vartheta_{d_{t-1}})$. The Dirichlet sufficient statistic vectors for the distributions over these parameters, $\alpha_{d,i}$ and α_d, have dimensionality equal to the number of possible states in modality i and the number of actions N respectively. They will effectively represent the number of observations of each action-modality-state and action-action pair. (So, for example, if we want a policy where camera 1 is shown preferentially when faces are visible, we could have $\alpha_{d=cam1,i=faces,s=face} > \alpha_{d=cam1,i=faces,s=no\,face}$.) The next action is selected by computing and then drawing from the multinomial predictive distribution over actions given the past examples $H_t = \{d_{1:t-1}, \mathbf{s}_{1:t-1}\}$ as follows,

$$\begin{aligned} p(d_t|\mathbf{s}_t,H_t) &= \int p(d_t,\theta|\mathbf{s}_t,H_t)d\theta, \\ &\propto \prod_i \int p(s_{i,t}|d_t,\theta_{d_t,i})p(\theta_{d_t,i}|H_t)d\theta_{d_t,i} \int p(d_t|d_{t-1},\vartheta_{d_{t-1}})p(\theta_{d_{t-1}}|H_t)d\vartheta_{d_{t-1}}, \\ &\propto \prod_i \text{Multi}(s_{i,t};\frac{\alpha_{d_t,i}}{\sum\alpha_{d_t,i}}) \cdot \text{Multi}(d_t;\frac{\alpha_{d_{t-1}}}{\sum\alpha_{d_{t-1}}}), \end{aligned} \quad (8)$$

To perform learning, after validation at each t, we update the posterior distribution over parameters θ given the new observations $\mathbf{s}_t, d_t$ as in Eq. 9,

$$\begin{aligned} p(\theta|\mathbf{s}_t,d_t,H_t) &\propto p(\mathbf{s}_t|d_t,\theta)p(\theta|H_t) \\ &= \prod_d\prod_i \text{Dir}(\theta_{d,i}|\hat{\alpha}_{d,i}) \\ \hat{\alpha}_{d,i,s} &= \alpha_{d,i,s} + I[s=s_{i,t}]I[d=d_t]. \end{aligned} \quad (9)$$

Assuming a factored prior distribution $p(\theta|H_t)$ the posterior also factorizes and learning simply requires incrementing appropriate elements of the policy sufficient statistic vectors α to reflect the new data. The transition model is learned similarly. Effectively, there are MN observation "experts" and one transition expert voting for each decision. Each expert's vote is based on the historical statistics of the observed modality and actions for which it is responsible.

As for the continuous case, we might wish to account for the potentially unequal informativeness of explicit intervention and implicit consent. For implicit learning purposes we would ideally model making a noisy observation of the true perfect decision, but this correlates $\theta_{d,i}$s and results in a non-factorized posterior over θ. A compromise is to pretend that the uncertainty during implicit consent is instead in the observations, now $\mathbf{s}'_t$. Integrating over the "unknown true" observations $\mathbf{s}_t$ given a noise model $p(\mathbf{s}'_t|\mathbf{s}_t,\varepsilon)$ brings in asymmetric noise in a tractable way, so Eq. 9 is replaced with:

$$\begin{aligned}
p(\theta|\mathbf{s}'_t,d_t,H_t) &\propto \sum_{\mathbf{s}_t}\prod_i p(s'_{i,t}|s_{i,t},\varepsilon)p(s_{i,t}|d_t,\theta_{d_t,i})p(\theta|H_t), \\
&\propto \prod_i\left(\sum_{s_i}\mathrm{Dir}(\theta_{d_t,i}|\hat{\alpha}_{d_t,i})p(s'_{i,t}|s_{i,t},\varepsilon)\right)\prod_{d:d\neq d_t}\prod_i\mathrm{Dir}(\theta_{d,i}|\alpha_{d,i}), \\
\hat{\alpha}_{d_t,i,s} &= \alpha_{d_t,i,s}+I[s=s_{t,i}]. \qquad (10)\\
p(\theta|\mathbf{s}'_t,d_t,H_t) &\simeq \prod_d\prod_i\mathrm{Dir}(\theta_{d,i}|\tilde{\alpha}_{d,i,s}). \qquad (11)
\end{aligned}$$

The posterior over policy θ again factorizes into a product (over decisions i and modalities d) (Eq. 11). Each policy factor $\theta_{d,i}$ is a sum over the updated policies given each possible observation s_i weighted by the likelihood $p(s'_i|s_i,\varepsilon)$. This can be approximated efficiently and accurately using moment matching[4] to get the statistics $\tilde{\alpha}$ of the final posterior factors. The empirical effect of this implicit consent learning is to add an increment $\kappa<1$ to the counter $\alpha_{d,i,s}$ for each observation $(d_t,s_{i,t})$ where κ decreases as more evidence is accumulated. Like the continuous case, ε can be set empirically. Implicit observations therefore count for less than explicit ones (where $\kappa=1$), and the policy confidence attainable purely by implicit consent is limited.

4.1 Results

Meeting Scenario We now apply the discrete action model to broadcast direction of a meeting recorded by multiple cameras. The aim here is to select an appropriate single view to broadcast at each time step. We use data from the AMI project corpus[1] as a source of raw multimedia meeting data from which five camera views (one of each of four participants and an overhead view) and the audio input from the four participant's lapel microphones are taken as input. To identify speech from background for each participant, the raw audio data is pre-processed into a binary speech activity feature by training a two component mixture of Gaussians on the signal power. As in the continuous case, face and motion presence features are also included in $\mathbf{s}_t$, which is therefore a twelve-dimensional binary vector. Based on these features, the model will learn a direction policy θ to specify d_t: how to switch the cameras over time.

Results from a meeting scenario are shown in Fig. 4. In this case, the model was trained during the first three minutes of viewing, requiring 32 interventions (see Fig. 4d). The next 60 seconds of data are illustrated by speech and motion activity plots in Fig. 4a. On the basis of these and the face features, the model chose the actions illustrated by Fig. 4b, for which sample frames are shown in Fig. 4c. In this case the model has learned the irrelevance of face features and the saliency of speech and motion. One challenge in this task is dealing with transitions between saliency of participants. As a person takes

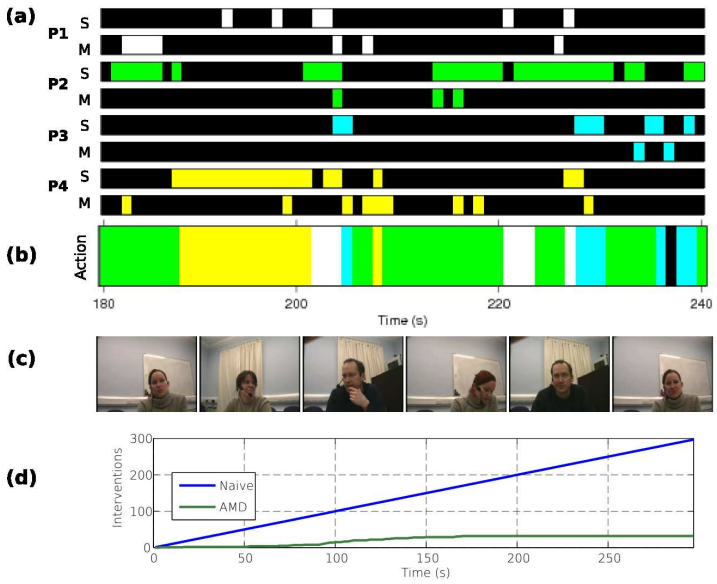

Figure 4: Meeting Scenario. (a) Input speech and motion activity features for participants 1-4. s - speech, m - motion. Shading indicates user activity. (b) Broadcast decision. Shading indicates the user broadcast. (c) Selected sample frames. (d) Training performance.

over the floor we might like the camera view to switch promptly. However, if two people are vying for the floor, speaking over each other, we might like periodically alternating shots of them rather than rapid oscillations of view. The structure of our model allows just this; cameras effectively have "experts" arguing for them on basis of the relevant input features, while the transition "expert" simply wants smooth variation. Therefore when one person takes over speaking, (e.g. at 190s) the experts agree and out-vote the transition model for a rapid response. When two people talk over each other, (e.g. 200-210s), the experts disagree, so sampling from the decision posterior without filtering would result in rapidly oscillating views of P1 or P2. However, in conjunction with the learned transition model, sampling from the decision posterior results in a salient yet fairly smoothly varying view. Finally if everyone or no-one is talking (e.g. around 235s), all the experts disagree and the chance of switching to the backup overview shot becomes significant.

4.2 Summary

In this section we described a class of machine direction problems and solutions where the learning of a direction policy corresponds to learning a dynamic classification model. We illustrated that the policy for a meeting broadcast problem can be learned online with only a small number of explicit instructions. The simple factored formulation reduces the number of parameters to learn and produces a model with easily interpretable behavior. At the same time, it is sufficiently flexible to learn new scenario or user preference policies. For example, a viewer may prefer to see responses of strong approval or disapproval of the speaker than to see the speaker herself. Although we do not attempt to do this

here (as computing emotion from speech and facial expression is currently expensive and unreliable) such features could trivially be included in our framework.

5 Discussion

Summary Machine direction systems have previously been engineered for various specific tasks. In this paper, we developed probabilistic models for this entire class of tasks, illustrating their underlying commonality as dynamic regression and classification problems. We introduced the novel task of learning such models from data online, and presented a real-time solution. This allows broadcast direction policies to be learned for novel scenarios, for which expert human or machine directors may not exist or be economical. Using our framework, it is possible to learn any new scenario if the input feature bank provides at least some relevant feature(s) (which need not be known in advance). We illustrated this by learning a novel scenario involving dance. Moreover, the nature of the problem allows policies to be learned with minimal user effort. This framework therefore allows individual user preferences to be learned, potentially enabling new patterns of future media consumption.

An alternative theoretical approach that could be used to model the AMD scenario is that of reinforcement learning (RL), in which the user simply rewards or punishes the model for its decisions. We did not pursue this because the impoverished nature of the feedback in RL means that much more training is needed to attain good performance. Instead, we exploit the opportunity to learn rapidly from few targeted explicit supervisions.

Future Work While the size and complexity of problems illustrated so far are limited, our framework is designed to be extensible. Adding further features for new problem scenarios is trivial and non-linear decision functions can be learned by changing the basis functions used or by using more sophisticated classifiers. We are also investigating more sophisticated temporal correlation modeling for improving response latency, smoothness and long range correlation modeling. Finally, we would also like to unify the different model forms currently used for discrete and continuous direction.

References

[1] M. Al-Hames, B. Hörnler, C. Scheuermann, and G. Rigoll. Using audio, visual, and lexical features in a multi-modal virtual meeting director. In *MLMI 2006, 3rd Joint Workshop on Multimodal Interaction and Related Machine Learning Algorithms*, 2006.

[2] D. Arijon. *Grammar of the film language*. Silman-James Press, 1991.

[3] Y. Ariki, S. Kubota, and M. Kumano. Automatic production system of soccer sports video by digital camera work based on situation recognition. In *Proceedings of the Eighth IEEE International Symposium on Multimedia (ISM '06)*, 2006.

[4] T. Minka. Divergence measures and message passing. Technical report, Microsoft Research, 2005.

[5] P. Viola and M. J. Jones. Robust real-time face detection. *IJCV*, 57:137–154, 2004.

[6] P. Yin, A. Criminisi, J. Winn, and I. Essa. Tree-based classifiers for bilayer video segmentation. In *Proc. IEEE Computer Vision and Pattern Recognition (CVPR)*, 2007.

[7] C. Zhang, Y. Rui, J. Crawford, and L.-W. He. An automated end-to-end lecture capture and broadcast system. *ACM Transacations on Multimedia Computing, Communications and Applications*, 4:1–23, 2008.