Toward Augmented Control Systems

Alex Pentland
Massachusetts Institute of Technology
Room E15-387, 20 Ames Street
Cambridge, MA 02139, USA
Tel (+1) 617-253-0648
Fax (+1) 617-253-8874
sandy@media.mit.edu

Andrew Liu
Nissan Cambridge Basic Research
4 Cambridge Center
Cambridge, MA, 02142
Tel (+1) 617-374-9669
Fax (+1) 617-374-9697
andy@pathfinder.cbr.com

Abstract

We describe research toward building control systems that include a complex, multi-state model of the driver's behavior. This can allow us to create a control system that augments, rather than replaces, the driver. To accomplish this requires inferring the internal state of the driver (preparing to brake, turn, etc.), and then correctly adapting the remainder of the control system to achieve optimal performance.

1 Introduction

Several automobile manufacturers have begun to experiment with "augmented control" systems [3, 4]. Examples are cruise control systems that help the driver maintain inter-vehicle distance as well as speed, and steering systems that help the driver maintain lane position. Such Augmented Control (AC) systems are fundamentally different than automated driving systems, because they complement rather than replace the driver. Consequently, they must consider the driver as an integral part of the control system, rather than as an external source of input.

The difficulty, of course, is that we generally do not understand humans well enough to construct a mathematical model of their input/output relations. This difficulty has forced most attempts at AC systems to adopt the simplest possible model of the human. For instance, if one models the human as a trivial "black box" such as a noise source or filter it is relatively easy to obtain a mathematical specification of the vehicle control system, including the human. Such simple models, however, also make it impossible to build a system that takes real advantage of the human's abilities.

Our approach is to instead model the human as a Markov device with a (possibly large) number of internal mental states, each with its own particular control behavior, and inter-state transition probabilities. A simple example of this type of human model would be a bank of standard quadratic controllers, each using different dynamics and measurements, together with a network of probabilistic transitions between them.

To integrate this human model into the overall AC system it is necessary to know which controller is currently "in charge," so that the remainder of the system (the car) can configure itself to achieve optimum overall performance. However the internal states of the human are not directly observable, so they must be determined through an indirect estimation process. One efficient and robust method of accomplishing this is to use the expectation-maximization method common in Hidden Markov Modeling (HMM).

By using these methods to infer the driver's internal state (e.g., are they passing, following, turning, etc.) it seems likely that we can design systems that are able to dynamically reconfigure themselves to better fit the situation. This can potentially allow for higher performance than is possible with a fixed model of the human (assuming similar controller complexity).

2 Static Models

The simplest non-trivial driver models that have been considered are static finite state machines. The driver is modeled as a set of states S, and there are certain legal transtitions between these states. For instance, a driver model might have only two states, braking and non-braking.

Helander [2] used this model in analyzing driver braking behavior. His data show that drivers exhibit an elevated electrodermal response (EDR) in their hands that reliably preceded braking. He suggested that by use of a suitably instrumented steering wheel the car's braking response could be improved.

3 Dynamic Models

Models that rely on "oracle" observations to indicate state are often unsatisfactory, usually because the external observable is not completely reliable and requires the use of extra sensors. Instead, we would like a driver model that analyzes the driver's control input to identify the state of the driver. Analysis of such data usually requires a dynamic model of the driver.

The simplest such driver model is a single dynamic process

$$\mathbf{X}_{k+1} = \mathbf{f}(\mathbf{X}_k, \Delta t) + \xi(t) \qquad (1)$$

where the function $\mathbf{f}$ models the dynamic evolution of state vector $\mathbf{X}_k$ at time k, and let us define an observation process

$$\mathbf{Y}_k = \mathbf{h}(\mathbf{X}_k, \Delta t) + \eta(t) \qquad (2)$$

where the sensor observations $\mathbf{Y}$ are a function $\mathbf{h}$ of the state vector and time. Both ξ and η are white noise processes having known spectral density matrices.

Using Kalman's result, we can then obtain the optimal linear estimate $\hat{\mathbf{X}}_k$ of the state vector $\mathbf{X}_k$ by use of the following *Kalman filter*:

$$\hat{\mathbf{X}}_k = \mathbf{X}_k^* + \mathbf{K}_k(\mathbf{Y}_k - \mathbf{h}(\mathbf{X}_k^*, t)) \qquad (3)$$

provided that the Kalman gain matrix $\mathbf{K}_k$ is chosen correctly [14]. At each time step k, the filter algorithm uses a state prediction $\mathbf{X}_k^*$, an error covariance matrix prediction $\mathbf{P}_k^*$, and a sensor measurement $\mathbf{Y}_k$ to determine an optimal linear state estimate $\hat{\mathbf{X}}_k$, error covariance matrix estimate $\hat{\mathbf{P}}_k$, and predictions $\mathbf{X}_{k+1}^*$, $\mathbf{P}_{k+1}^*$ for the next time step.

The prediction of the state vector $\mathbf{X}_{k+1}^*$ at the next time step is obtained by combining the optimal state estimate $\hat{\mathbf{X}}_k$ and Equation 1:

$$\mathbf{X}_{k+1}^* = \hat{\mathbf{X}}_k + \mathbf{f}(\hat{\mathbf{X}}_k, \Delta t)\Delta t \qquad (4)$$

In our application this prediction equation is also used with larger times steps, to predict the driver's future state. This prediction allows us to maintain synchrony with the driver by giving us the lead time needed to alter suspension components, etc.

Finally, given the state vector $\mathbf{X}_k$ at time k we can predict the measurements at time $k + \Delta t$ by

$$\mathbf{Y}_{k+\Delta t} = \mathbf{h}(\mathbf{X}_k, \Delta t) \qquad (5)$$

and the predicted state vector at time $k + \Delta t$ is given by

$$\hat{\mathbf{X}}_{k+\Delta t} = \mathbf{X}_k^* + \mathbf{f}(\hat{\mathbf{X}}_k, \Delta t)\Delta t \qquad (6)$$

3.1 Multiple Dynamic Models

Driver behavior, of course, is not as simple as a single dynamic model. The next most complex model of driving behavior is to have *several* alternative models of the driver's dynamics, one for each class of response. Then at each instant we can make observations of the driver's state, decide which model applies, and then make our response based on that model. This is known as the *multiple model* or *generalized likelihood* approach, and produces a generalized maximum likelihood estimate of the current and future values of the state variables [15]. Moreover, the cost of the Kalman filter calculations is sufficiently small to make the approach quite practical.

Intuitively, this solution breaks the driver's overall behavior down into several "prototypical" behaviors. For instance, we might have dynamic models corresponding to a relaxed driver, a very "tight" driver, and so forth. We then classify the driver's behavior by determining which model best fits the driver's observed behavior.

Mathematically, this is accomplished by setting up a set S of Kalman filters, one for the dynamics of each model:

$$\hat{\mathbf{X}}_k^{(i)} = \mathbf{X}_k^{*(i)} + \mathbf{K}_k^{(i)}(\mathbf{Y}_k - \mathbf{h}^{(i)}(\mathbf{X}_k^{*(i)}, t)) \qquad (7)$$

where the superscript (i) denotes the i^{th} Kalman filter. The *measurement innovations process* for the i^{th} model (and associated Kalman filter) is then

$$\Gamma_k^{(i)} = \mathbf{Y}_k - \mathbf{h}^{(i)}(\mathbf{X}_k^{*(i)}, t) \qquad (8)$$

The measurement innovations process is zero-mean with covariance $\mathcal{R}$.

The i^{th} measurement innovations process is, intuitively, the part of the observation data that is unexplained by the i^{th} model. The model that explains the largest portion of the observations is, of course, the model most likely to be correct. Thus, at each time step, we calculate the probability $Pr^{(i)}$ of the m-dimensional observations $\mathbf{Y}_k$ given the i^{th} model's dynamics,

$$Pr^{(i)}(\mathbf{Y}_k) = \frac{\exp\left(-\frac{1}{2}\Gamma_k^{(i)T}\mathcal{R}^{-1}\Gamma_k^{(i)}\right)}{(2\pi)^{m/2}\mathrm{Det}(\mathcal{R})^{1/2}} \qquad (9)$$

and choose the model with the largest probability. This model is then used to estimate the current value of the state variables, to predict their future values, and to choose among alternative responses.

Note that when optimizing predictions of measurements Δt in the future, Equation 8 must be modified slightly to test the predictive accuracy of state estimates from Δt in the past.

$$\Gamma_k^{(i)} = \mathbf{Y}_k - \mathbf{h}^{(i)}(\mathbf{X}_{k-\Delta t}^{*(i)} + \mathbf{f}^{(i)}(\hat{\mathbf{X}}_{k-\Delta t}^{(i)}, \Delta t)\Delta t, t)) \tag{10}$$

by substituting Equation 6.

3.2 Results

We have used this method to accurately remove lag in a high-speed telemanipulation task by continuously re-estimating the user's arm dynamics (e.g., tense and stiff, versus relaxed and inertia-dominated) [5].

In this case, the state vector $\mathbf{X}_k$ consists of the true position, velocity, and acceleration of the hand in each of the x, y, and z coordinates, and the observation vector $\mathbf{Y}_k$ consists of the position readings for the x, y, and z coordinates. We found that using this multiple-model approach we were able to obtain significantly better predictions of the user's hand position that was possible using a single dynamic or static model.

4 Hidden Markov Modeling

In the above multiple dynamic model, all the processes have a fixed likelihood at each time step. However, this is uncharacteristic of most driving situations, where there is a fixed sequence of internal states each with its own dynamics. Consider driving through a curve; the driver may be modeled as having transitioned through a series of states $\lambda = (s_1, s_2, ...s_k), s_i \epsilon S$, for instance, entering a curve, in the curve, and exiting a curve, and other. Transitions between these states happened only in the order indicated, with a final transition from other to entering the curve.

Thus in considering state transitions among a set of dynamic models we should make use of our current estimate of the driver's internal state. We can accomplish this fairly generally by considering the Markov probability structure of the transitions between the different states. The input to decide the driver's current state (e.g., which dynamic model currently applies) will be the measurement innovations process as above, but instead of using this directly in Equation 9 we will instead consider the Markov inter-state transition probabilities.

While a substantial body of literature exists on HMM technology [7, 8, 10, 13], we will first briefly outline a traditional discussion of the algorithms. After outlining the fundamental theory in training and testing of a discrete HMM, we will generalize these results to the continuous density case applicable to switching between dynamic models. For broader discussion of the topic, [8, 11] are recommended.

A time domain process demonstrates a Markov property if the conditional probability density of the current event, given all present and past events, depends only on the j^{th} most recent events. If the current event depends solely on the most recent past event, then the process is a first order Markov process.

The initial topology for an HMM can be determined by estimating how many different states are involved in the observed phenominon. Fine tuning this topology can be performed empirically. Figure 1, for instance, shows a four state HMM with skip transitions that we have used to classify complex hand motions.

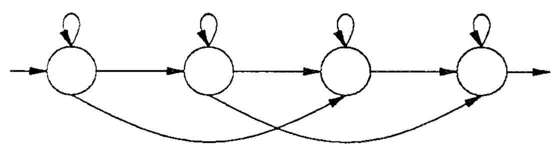

Figure 1: The four state HMM used for recognition, from [6].

There are three key problems in HMM use. These are the evaluation, estimation, and the decoding problems. The evaluation problem is that given an observation sequence and a model, what is the probability that the observed sequence was generated by the model ($Pr(\mathbf{Y}|\lambda)$) (notational style adapted from [8])? If this can be evaluated for all competing models for an observation sequence, then the model with the highest probability can be chosen for recognition.

$Pr(\mathbf{Y}|\lambda)$ can be calculated several ways. The naive way is to sum the probability over all the possible state sequences in a model for the observation sequence:

$$Pr(\mathbf{Y}|\lambda) = \sum_{i \epsilon S} \prod_{t=1}^{k} a_{t-1}(i) b_i(Y_t) \tag{11}$$

where $a_k(i)$ are the state transition probabilities, and $b_k(Y)$ are the output probabilites.

However, this method is exponential in time, so the more efficient forward-backward algorithm is used in practice. The following algorithm defines the forward variable α and uses it to generate $Pr(\mathbf{Y}|\lambda)$,

- $\alpha_1(i) = \pi_i b_i(O_1)$, for all states i, where π_i are the initial state probabilities (by default we can let $\pi_i = \frac{1}{n_I}$),
- Calculating $\alpha()$ along the time axis, for $t = 2, ..., k$, and all states j, compute

$$\alpha_t(j) = [\sum_i \alpha_{t-1}(i) a_{ij}] b_j(Y_t) \quad (12)$$

- Final probability is given by

$$Pr(\mathbf{Y}|\lambda) = \sum_{i \epsilon S} \alpha_k(i) \quad (13)$$

The first step initializes the forward variable with the initial probability for all states, while the second step inductively steps the forward variable through time. The final step gives the desired result $Pr(\mathbf{Y}|\lambda)$, and it can be shown by constructing a lattice of states and transitions through time that the computation is only order $O(N^2T)$. The backward algorithm, using a process similar to the above, can also be used to compute $Pr(\mathbf{Y}|\lambda)$.

The estimation problem concerns how to adjust λ to maximize $Pr(\mathbf{Y}|\lambda)$ given an observation sequence $\mathbf{Y}$. Given an initial model, which can have flat probabilities, the forward-backward algorithm allows us to evaluate this probability. All that remains is to find a method to improve the initial model. Unfortunately, an analytical solution is not known, but an iterative technique can be employed.

Using the actual evidence from the training data, a new estimate for the respective output probability can be assigned:

$$\bar{b}_j(Y) = \frac{\sum_{t \epsilon (Y_t = Y)} \gamma_t(j)}{\sum_{t=1}^{k} \gamma_t(j)} \quad (14)$$

where $\gamma_t(i)$ is defined as the posterior probability of being in state i at time t given the observation sequence and the model. Similarly, the evidence can be used to develop a new estimate of the probability of a state transition ($\bar{a}_{ij}$) and initial state probabilities ($\bar{\pi}_i$).

Thus all the components of model (λ) can be reestimated. Since either the forward or backward algorithm can be used to evaluate $Pr(\mathbf{Y}|\bar{\lambda})$ versus the previous estimation, the above technique can be used iteratively to converge the model to within some error criterion. While the technique described only handles a single observation sequence, it is easy to extend to a set of observation sequences. A more formal discussion can be found in [7, 8, 13].

While the estimation and evaluation processes described above are sufficient for the development of an HMM system, the Viterbi algorithm provides a quick means of evaluating a set of HMM's in practice as well as providing a solution for the decoding problem. In decoding, the goal is to recover the state sequence given an observation sequence. The Viterbi algorithm can be viewed as a special form of the forward-backward algorithm where only the maximum path at each time step is taken instead of all paths. This optimization reduces computational load and additionally allows the recovery of the most likely state sequence. The steps to the Viterbi are

- Initialization. For all states i, $\delta_1(i) = \pi_i b_i(Y_1)$; $\psi_i(i) = 0$
- Recursion. From $t = 2$ to k and for all states j, $\delta_t(j) = \max_i[\delta_{t-1}(i) a_{ij}] b_j(Y_t)$; $\psi_t(j) = \text{argmax}_i[\delta_{t-1}(i) a_{ij}]$
- Termination. $P = \max_{s \epsilon S}[\delta_k(s)]$; $s_k = \text{argmax}_{s \epsilon S}[\delta_k(s)]$
- Recovering the state sequence. From $t = k-1$ to 1, $s_t = \psi_{t+1}(s_{t+1})$

Note that since Viterbi only guarantees the maximum of $Pr(\mathbf{Y}, S|\lambda)$ over all state sequences S (as a result of the first order Markov assumption) instead of the *sum* over all possible state sequences, the resultant scores are only an approximation. However, [10] shows that this is often sufficient.

4.1 The Continuous Case

So far this discussion of HMMs has assumed some sort of quantization of feature vectors into classes. However, instead of using vector quantization, the actual probability densities for the features may be used. Baum-Welch, Viterbi, and the forward-backward algorithms can be modified to handle a variety of characteristic densities [9]. In this context, however, the densities will be assumed to be Gaussian. Specifically, from Equation 9,

$$b_j(Y_t) = \frac{\exp\left(-\frac{1}{2}\Gamma_k^{(i)T}\mathcal{R}^{-1}\Gamma_k^{(i)}\right)}{(2\pi)^{m/2}\mathrm{Det}(\mathcal{R})^{1/2}} \qquad (15)$$

Initial estimations of μ and σ may be calculated by dividing the evidence evenly among the states of the model and calculating the mean and variance in the normal way. Whereas flat densities were used for the initialization step before, the evidence is used here. Now all that is needed is a way to provide new estimates for the output probability. This can be accomplished by the Kalman filter update equations.

4.2 Results

We have used this method, albeit with only the simplest of dynamic models, to interpret and classify a set of forty complex, two-hand motions [6]. The motions were continuous and showed severe co-articulation effects. Input descriptions were hand position, orientation, and aspect ratio. We were able to obtain 99.2% accuracy at this classification task.

5 Work in Progress

We are now using this approach to use observations of driver head, hand, and leg movements to classify the driver's internal state. We are currently investigating results from Land [1] suggesting that a 4 state HMM with simple dynamic models may be sufficient to use driver head position to predict when a driver is preparing to enter or leave a curve, and to estimate the change in steering angle. The ability to accurately predict such control input from observations of driver head position may make it possible (for instance) to dyamically tune the vehicle suspension.

This research is being conducted within the Nissan Cambridge Basic Research driving simulator. The simulator consists of the front half of a Nissan 240SX convertible and a 60 deg (horizontal) by 40 deg (vertical) image projected onto the wall facing the driver. The 240SX is instrumented to record driver control input such as steering wheel angle, brake position, and accelerator position. Head and eye positions can be measured simultaneously using the ISCAN HeadHunter Eye Tracking System. A timing pulse is sent over a serial connection from the simulation computer to the eye tracking computer to synchronize the data for the off-line analysis.

We arecurrently simulating driving situations such as driving at a constant speed on a smoothly curving road with no traffic, and maintaining the distance behind a lead car moving at a constant speed along the same curvy route. The head and eye position data and driver input data from these "prototypical" driving situations are then used to develop the dynamic models of driver behavior for a specific driving state. For instance, in the first driving situation, initial results show a predictive relationship between driver head angle and steering wheel movement. We believe that similar relationships may also be obtained in other experimental conditions. The results from these initial experiments will be reported at the conference.

Acknowledgements. We would particularly like to thank Thad Starner for his work in developing HMM technology for gesture understanding.

References

[1] M. Land. Predictable eye-head coordination during driving, *Nature*, Vol. 359, pp. 318-320, 1992.

[2] M. Helander. Applicability of Drivers' Electrodermal Response to the Design of the Traffic Environment, *J. Applied Psychology*, Vol. 63, No. 4, pp. 481-488, 1978.

[3] S. Tsugawa, M. Aoki, H. Akio, and K. Seki. Recent Japanese Projects of AVCS-Related Systems, *IEEE Intelligent Vehicles Symposium 94*, pp. 125-130, Paris, France, June 20-24, 1994.

[4] U. Franke, S. Mehring, A. Suissa, and S. Hahn. The Daimler-Benz Steering Assistant – a Spin-off from Autonomous Driving, *IEEE Intelligent Vehicles Symposium 94*, pp. 120-124, Paris, France, June 20-24, 1994.

[5] M. Friedmann, T. Starner, and A. Pentland. Device Synchronization using an Optimal Linear Filter, *Proc. ACM 1992 Symposium on Interactive 3D Graphics*, Boston, MA, May 1992.

[6] T. Starner, and A. Pentland. Visual Recognition of American Sign Language Using Hidden Markov Models, *Proc. Int'l Workshop on Automatic Face- and Gesture-Recognition*, Zurich, Switzerland, June 26-28, 1995.

[7] L. Baum. An inequality and associated maximization technique in statistical estimation of probabilistic functions of markov processes. *Inequalities*, 3:1–8, 1972.

[8] X. Huang, Y. Ariki, and M. Jack. *Hidden Markov Models for Speech Recognition*. Edinburgh Univ. Press, Edinburgh, 1990.

[9] B. Juang. Maximum likelihood estimation for mixture multivariate observations of markov chains. *AT&T Technical Journal*, 64:1235–1249, 1985.

[10] L. Rabiner and B. Juang. An introduction to hidden markov models. *IEEE ASSP Magazine*, p. 4–16, Jan. 1996.

[11] T. Starner. Visual Recognition of American Sign Language Using Hidden Markov Models. Master's thesis, MIT Media Laboratory, Feb. 1995.

[12] T. Starner, J. Makhoul, R. Schwartz, and G. Chou. On-line cursive handwriting recognition using speech recognition methods. In *ICASSP*, 1994.

[13] S. Young. *HTK: Hidden Markov Model Toolkit V1.5*. Cambridge Univ. Eng. Dept. Speech Group and Entropic Research Lab. Inc., Washington DC, Dec. 1993.

[14] R.E. Kalman and R.S. Bucy. New results in linear filtering and prediction theory. In *Transaction ASME (Journal of basic engineering)*, 83D, 95-108, 1961.

[15] A.S. Willsky. Detection of Abrupt Changes in Dynamic Systems. In M. Basseville and A. Benveniste, (Ed.). *Detection of Abrupt Changes in Signals and Dynamical Systems*, Lecture Notes in Control and Information Sciences, No. 77, pp. 27-49, Springer-Verlag, 1986.