Modeling Multiple-mode Systems with Predictive State Representations

Britton Wolfe
Computer Science
Indiana University-Purdue
University Fort Wayne
wolfeb@ipfw.edu

Michael R. James
AI and Robotics Group
Toyota Technical Center
michael.r.james@gmail.com

Satinder Singh
Computer Science and
Engineering
University of Michigan
baveja@umich.edu

***Abstract*— Predictive state representations (PSRs) are a class of models that represent the state of a dynamical system as a set of predictions about future events. This work introduces a class of structured PSR models called multi-mode PSRs (MMPSRs), which were inspired by the problem of modeling traffic. In general, MMPSRs can model uncontrolled dynamical systems that switch between several modes of operation. An important aspect of the model is that the modes must be recognizable from a window of past *and future* observations. Allowing modes to depend upon future observations means the MMPSR can model systems where the mode cannot be determined from only the past observations. Requiring modes to be defined in terms of observations makes the MMPSR different from hierarchical latent-variable based models. This difference is significant for learning the MMPSR, because there is no need for costly estimation of the modes in the training data: their true values are known from the mode definitions. Furthermore, the MMPSR exploits the modes' recognizability by adjusting its state values to reflect the true modes of the past as they become revealed. Our empirical evaluation of the MMPSR shows that the accuracy of a learned MMPSR model compares favorably with other learned models in predicting both simulated and real-world highway traffic.**

***Index Terms*— highway traffic, predictive state representations, dynamical systems**

I. Multi-mode PSRs (MMPSRs)

Predictive state representations (PSRs) [3] are a class of models that represent the state of a dynamical system as a set of predictions about future events. PSRs are capable of representing partially observable, stochastic dynamical systems, including any system that can be modeled by a finite partially observable Markov decision process (POMDP) [5]. There is evidence that predictive state is useful for generalization [4] and helps to learn more accurate models than the state representation of a POMDP [7]. This work introduces a class of structured hierarchical PSR models called multi-mode PSRs (MMPSRs) for modeling uncontrolled dynamical systems that switch between several modes of operation. Unlike latent-variable models like hierarchical HMMs [2], the MMPSR requires that the modes be a function of past and future observations. This requirement yields advantages both when learning and using an MMPSR, as explained throughout this section.

The MMPSR is inspired by the problem of predicting cars' movements on a highway. One way to predict the car's movements would be to determine what mode of behavior the car was in — e.g., a left lane change, right lane change, or going straight — and make predictions about the car's movement conditioned upon that mode of behavior. The MMPSR makes predictions in this way using two component models which form a simple, two-level hierarchy (Figure 1). When modeling highway traffic, the high-level model will predict the mode of behavior, and the low-level model will make predictions about the car's future positions conditioned upon the mode of behavior. The remainder of this section formalizes the MMPSR model in general terms, making it applicable to dynamical systems other than highway traffic.

A. Observations and Modes

The MMPSR can model uncontrolled, discrete-time dynamical systems, where the agent receives some observation O_i at each time step $i = 1, 2, \ldots$. The observations can be vector-valued and can be discrete or continuous. In addition to the observations that the agent receives from the dynamical system, the MMPSR requires that there exists a discrete set of *modes* the system could be in, and that there is some mode associated with each time step. The system can be in the same mode for several time steps, so a single mode can be associated with multiple contiguous time steps. The i^{th} mode since the beginning of time will be denoted by ψ_i, and $\psi(\tau)$ will denote the mode for the τ time step (Figure 2).

What distinguishes MMPSR models from hierarchical latent-variable models (e.g., hierarchical HMMs [2]) is the fact that the modes are not latent. Instead, they are defined in terms of past and possibly *future* observations. Specifically, the modes used by an MMPSR must satisfy the following *recognizability requirement*: There is some finite k such that, for any sequence of observations $O_1, \ldots, O_\tau, O_{\tau+1}, \ldots O_{\tau+k}$ (for any $\tau \geq 0$), the modes $\psi(1), \ldots, \psi(\tau-1), \psi(\tau)$ are known at time $\tau+k$ (or before). We say that a mode $\psi(\tau)$ is *known at time* τ' (where τ' can be greater than τ) if the definitions of the modes and the observations $O_1, \ldots, O_{\tau'}$ from the beginning of time through time τ' unambiguously determine the value of $\psi(\tau)$.

To reiterate, the recognizability requirement differentiates the MMPSR from hierarchical latent-variable models. If one were to incorporate the fact that modes were recognizable into a hierarchical latent-variable model, one would in effect get an MMPSR. The recognizability of the modes plays a crucial role in learning an MMPSR, because the modes for the batch of training data are known. If the modes were not recognizable, one would have to use the expectation-maximization algorithm to estimate the latent modes, as is

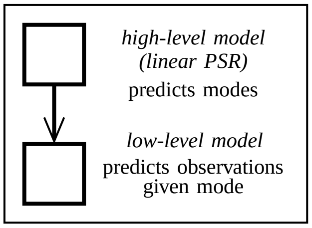

Fig. 1. The component models in an MMPSR.

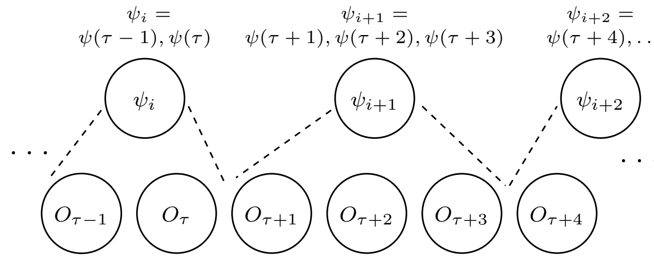

Fig. 2. How the mode variables relate to the observations. The observation at time τ is O_τ, the i^{th} mode seen since the beginning of time is ψ_i, and $\psi(\tau)$ is the mode at time τ.

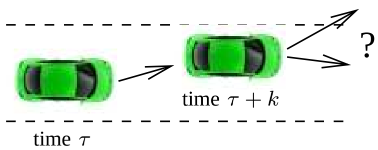

Fig. 3. A situation in the traffic system where the mode for the last few time steps of history is unknown. After moving from the position on the left of the figure at time τ to the position towards the right at time $\tau + k$, it is unclear if the car is beginning a left lane change or is just weaving in its lane. These two possibilities will assign different modes to the time steps τ through $\tau + k$ (i.e., "left lane change" vs. "going straight").

typical in hierarchical latent-variable models. The MMPSR also exploits the recognizability of the modes when maintaining its state, as described in Section I-B.

Because the modes can be defined in terms of past *and future* observations, the MMPSR is not limited to using history-based modes. A model using history-based modes would only apply to systems where the current mode $\psi(\tau)$ was always known at time τ. In contrast, the MMPSR can model systems where the agent will not generally know the mode $\psi(\tau)$ until several time steps after τ. The traffic system is an example system where there are natural modes (e.g. a "left lane change" mode) that can be defined in terms of past *and future* observations, but not past observations alone. During the first few time steps of a left lane change, the mode at those times is not known from the past observations of the car (Figure 3): the car could be in the "going straight" mode but weaving in its lane, or it could be starting the "left lane change" mode. Even though the "left lane change" mode will not immediately be known, it can be recognized when the car crosses the lane boundary. Thus, the "left lane change" mode can be defined as "The car crossed a lane boundary from right to left within the most recent k time steps *or it will do so* within the next k time steps."

Defining the modes in terms of future observations provides a common characteristic with other PSR models, where the state is defined in terms of future observations. The PSR literature shows that a handful of features of the short-term future can be very powerful, capturing information from arbitrarily far back in history [5]. This motivates the MMPSR's use of modes that are defined in terms of past and future observations.

The recognizability requirement is a constraint on the *definitions of the modes* and not on the dynamical system itself. That is, requiring the modes to be recognizable does *not* limit the dynamical systems one can model with an MMPSR. Given any system, one can define a recognizable set of modes by simply ensuring that the modes partition the possible history/length-k-future pairs (for some finite k). Then for any history and length-k future, exactly one mode definition is satisfied.

Furthermore, the modes that are defined for use by the MMPSR do not need to match the modes that the system actually used to generate the data (although a closer match might improve the accuracy of the MMPSR). This flexibility permits mode definitions that are approximations of complicated concepts like lane changes. Section II-A includes experiments where the mode definitions approximate the process used to generate the data.

In addition to the recognizability requirement, the MMPSR makes the following independence assumptions that characterize the relationship between modes and observations: (1) The observation $O_{\tau+1}$ is conditionally independent of the history of modes given the mode at time $\tau + 1$ and the history of observations $O_1, \ldots, O_\tau$. (2) The future modes $\psi_{i+1}, \ldots$ are conditionally independent of the observations through the end of ψ_i, given the history of modes $\psi_1, \ldots, \psi_i$. Even if the independence properties do not strictly hold, the MMPSR forms a viable approximate model, as demonstrated by the empirical results (Section II).

These independence properties lead to the forms of the high and low-level models within the MMPSR. The low-level model makes predictions for the next observation given the history of observations and the mode at the next time step (Figure 4.b). The low-level model will also predict the probability of a mode ending at time τ, given $\psi(\tau)$ and the history of observations through time τ. The high-level model makes predictions for future modes given the history of modes (Figure 4.a). Because of the second independence assumption, the high-level model can be learned and used independently from the low-level model; it models the sequence of modes $\psi_1, \psi_2, \ldots$ while abstracting away details of the observations.

B. Updating the MMPSR State

The state of the MMPSR must be updated every time step to reflect the new history. Suppose for a moment that for all τ, $\psi(\tau)$ was known at time τ. Then the high-level model would update its state whenever a mode ends, using that mode's value as its "observation." The low-level model would update its state after every time step τ, conditioned upon the most recent observation O_τ *and* the mode $\psi(\tau)$.

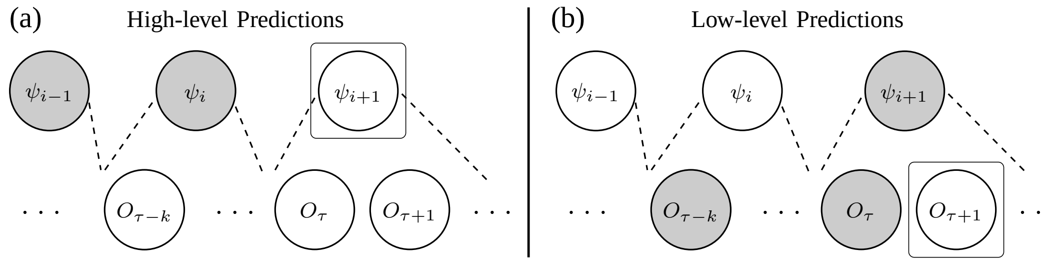

Fig. 4. Predictions made by the component models. The subscripts for the modes differ from the observations because each mode will last for several time steps (so $i << \tau$). (a) The high-level model predicts ψ_{i+1} given the history of modes (shaded). (b) The low-level model predicts $O_{\tau+1}$ given the history of observations and current mode (shaded).

Even though $\psi(\tau)$ will not always be known at time τ, this process defines the states of the high and low-level models under the assumption that some hypothetical values $\psi(1), \ldots, \psi(\tau)$ are the modes of history. Each sequence of hypothetical values has some posterior probability given the observations through time τ. The number of sequences $\psi(1), \ldots, \psi(\tau)$ with non-zero posterior probability will remain bounded, even as $\tau \rightarrow \infty$. Specifically, *only one sequence of hypothetical values for the known modes will have non-zero posterior probability* (i.e., the modes' true values), and only a finite (and typically small) window of past modes will be unknown, because of the recognizability requirement.

The MMPSR state at time τ consists of the posterior distribution over the modes of history and the high and low-level model states corresponding to each sequence $\psi(1), \ldots, \psi(\tau)$ with non-zero posterior probability. At the next time step $\tau + 1$, the MMPSR updates its state as follows. The MMPSR computes the high and low-level models' states for a given $\psi(1), \ldots, \psi(\tau), \psi(\tau + 1)$ from the high and low-level models' states at time τ using the respective model updates. The MMPSR updates the posterior distribution over modes of history using Bayes' Rule (see online appendix for details: `http://users.ipfw.edu/wolfeb/itsc2010appendix.pdf`).

Because the learned component models of an MMPSR will not be perfectly accurate, the Bayesian posterior update might not assign zero probability to hypothetical mode values even if they contradict the recognized value for those modes. Thus, in addition to the Bayesian posterior update, after each time step the MMPSR explicitly assigns zero probability to values of the history modes that contradict the known values, renormalizing the distribution after those changes. In addition, one can use pruning techniques (e.g. keep only the k most likely sequences of history modes) to reduce the number of history mode sequences that are maintained in the posterior distribution.

C. Making Predictions with an MMPSR

The MMPSR can make predictions about the next observation $O_{\tau+1}$ given the history of observations $O_1, \ldots, O_\tau$. Predictions of this form can be combined to make any prediction about the system, including predictions further than one step in the future. For a given assignment to the modes $\psi(1), \ldots, \psi(\tau+1)$, the low-level model can directly predict $O_{\tau+1}$ given $O_1, \ldots, O_\tau$. Since not all of $\psi(1), \ldots, \psi(\tau+1)$ will be known at time τ, the MMPSR makes predictions about $O_{\tau+1}$ given $O_1, \ldots, O_\tau$ by marginalizing the modes that are not known. This marginalization can be done relatively efficiently by using the distribution over modes that the MMPSR maintains.

To make predictions about observations several time steps in the future, it can be more efficient to make those predictions directly rather than calculating them from several next-observation predictions (cf. [5]). For example, in our empirical results, we include in the MMPSR several regression models to predict features of the future given the state of the low-level model and the most recent mode. As with the next-observation predictions, the overall prediction from the MMPSR marginalizes the unknown modes of history.

D. Learning an MMPSR

Learning an MMPSR consists of learning the high and low-level component models from a single sequence of observations (i.e., the training data). The high-level model is a linear PSR, so it is learned by applying the suffix-history algorithm [7] to the sequence of modes of the training data. The recognizability of the modes ensures that the learning algorithm can correctly and automatically determine the mode for each time step of the training data (except a few time steps at the beginning and/or end of the data).

Learning the low-level model also requires the correct modes of the training data, since the low-level model makes predictions and updates its state conditioned upon the corresponding mode. One way to implement this conditional form is to have separate parameters of the low-level model for each mode. For example, the low-level model could consist of several parameterized functions from features of history (i.e., the state of the low-level model) to predictions about the next observation, with a separate function for each mode. The function for each respective mode can then be learned by applying an appropriate regression method to the time steps of training data that have that mode. This form allows the low-level model to make specialized predictions for each mode. It is the form used in the following experiments.

II. Experiments

We learned MMPSR models for three systems: a simple random walk system, and both simulated and real-world highway traffic. The low-level model includes parameterized

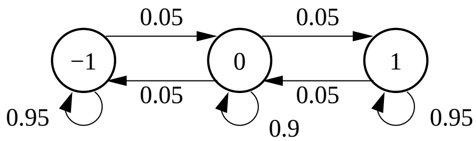

Fig. 5. The Markov chain to generate the modes for the random walk system.

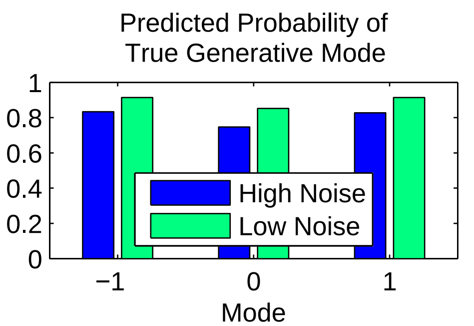

Fig. 6. Predicting the current generative mode in the random walk system with an MMPSR.

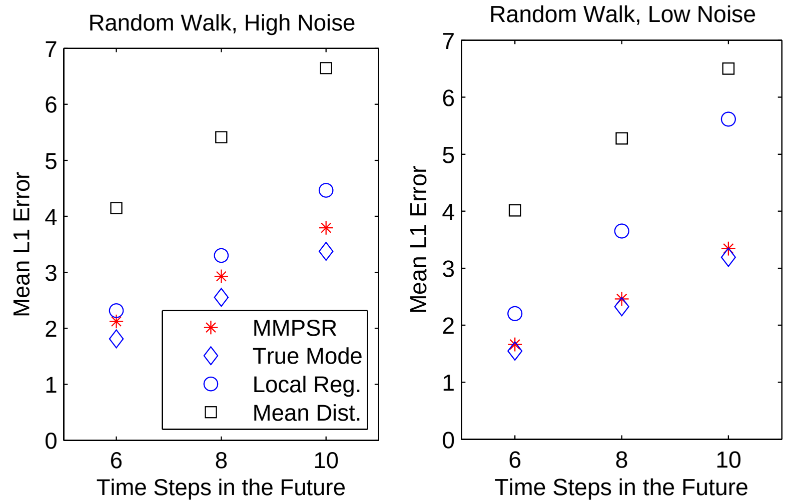

Fig. 7. Prediction error of the MMPSR and comparison models. To give a qualitative sense of the error, "Mean Dist." is the average distance traversed by the random walk. "True Mode" is the error when the MMPSR can artificially peer into the future to recognize the current mode. "Local Reg." uses locally-weighted linear regression over the entire data set.

functions for each mode that map features from a finite window of history to predictions of interest. Each function was learned using locally-weighted linear regression [1], a flexible non-linear function approximation method. The high-level linear PSR was learned using the suffix-history algorithm [7]. Because the modes are defined in terms of observations, the modes of the training data are known, which is critical in ensuring efficient training of the model. If the modes were not known, then an iterative estimation procedure would be needed to estimate the modes (e.g. expectation-maximization), often requiring several iterations to converge. At each iteration, the estimated modes would change, so the low-level model would have to be re-learned. In contrast, the low-level model of the MMPSR is only learned one time, using the true values of the modes because the modes are recognizable. For purposes of comparison, it would be possible to adapt hierarchical HMMs to exploit these modes, but as discussed above this would effectively yield an MMPSR. The comparison would then become a comparison between the suffix-history learning algorithm and the EM learning algorithm, a comparison which has already been done [7] with results showing the suffix-history algorithm to be generally superior. Therefore, we do not evaluate hierarchical HMMs here.

A. Random Walk

The empirical evaluation of the MMPSR begins with a simple random walk system, where the (scalar) observation at each time step is the change in (one-dimensional) position from the last time step. That change is given by the mode — which can take on values -1, 0, and 1 — plus mean-zero Gaussian noise. The mode is determined by a Markov chain, shown in Figure 5. The low-level model state consists of the average observations over the last k time steps for $k \in \{2, 5, 10\}$, along with a constant 1.0.

Because of the noise, the modes used to generate the data — which we call the *generative modes* — cannot be defined in terms of observations. Nevertheless, one can still define a set of *recognizable modes* in terms of observations. For the random walk system, the recognizable mode for time step τ is defined as the mode (-1, 0, or 1) that is closest to the average observation over five time steps, from $\tau - 2$ through $\tau + 2$. (Note that this mode definition includes both historical and future observations.)

While there will not be perfect correspondence between the recognizable modes and the generative modes, there is enough correlation that an MMPSR learned using the recognizable modes models the system well. Figure 6 shows the average likelihood that the MMPSR assigns (from its distribution over modes of history) to each *recognizable* mode whenever its related *generative* mode was the (latent) system mode at that time step. The high and low noise results correspond to using Gaussian noise with standard deviations 0.5 and 0.25, respectively. Though the higher noise leads to less certainty about the mode, the MMPSR attributes high probability to the generative modes for both noise levels.

In addition to tracking the modes well, the MMPSR is also able to predict the future observations well. The average error for predicting several time steps in the future is shown in Figure 7 for the high and low-noise systems. The MMPSR consistently achieves lower error than using locally-weighted linear regression on the entire data set, which does not explicitly leverage the existence of modes. For another comparison, we ran an additional set of experiments where we artificially allowed the MMPSR to peer into the future and determine the value of the current mode (even though the current mode is not generally recognizable from the observations through the current time). The real MMPSR (which does not peer into the future) does almost as well as the artificial MMPSR that can peer into the future, especially in the low-noise system (Figure 7).

2 sec.	L1 Error (feet)	Percent Error
MMPSR	4.856 ± 0.722	1.615 ± 0.219
Last Velocity	7.952 ± 0.130	3.397 ± 0.063
Local Regression	184.165 ± 16.022	94.298 ± 7.105

5 sec.	L1 Error (feet)	Percent Error
MMPSR	9.172 ± 1.950	1.151 ± 0.278
Last Velocity	19.197 ± 0.527	3.176 ± 0.058
Local Regression	420.237 ± 62.923	86.694 ± 11.396

Fig. 8. Error in predicting distance traveled for simulated traffic. The MMPSR achieves the lowest error for predicting both 2 and 5 seconds in the future. The confidence intervals are two standard deviations, computed across 15 data sets.

B. *Traffic*

We also learned MMPSRs to model simulated and real-world highway traffic. The MMPSR predicts the movements of a car given the history of that car and some neighboring cars. The observation at each time step consists of the x velocity, y velocity, and headway for the car being modeled, where headway is the time to collision with the car in front. We use "x" to refer to the lateral or side-to-side direction, while "y" refers to the longitudinal or forward direction. The features of history that composed the state of the low-level model were as follows: average y velocity over three different windows (the most recent 0.5 seconds, the 0.5 seconds before that, and the most recent 5 seconds); average x velocity over the most recent 0.5 seconds, and the 0.5 seconds before that; the x position; the distance to the closest car in front, and its average y velocity over the last 1.0 seconds; and a constant term. We used left and right lane changes as two of the modes of behavior for both simulated and real traffic; they were defined as any 4.0-second window where the car center crossed a lane boundary at the midpoint of the window.

1) Simulated Traffic: The simulated traffic consists of three lanes of traffic, with cars entering the field of view stochastically. (A video is available at `http://www.youtube.com/watch?v=8eHx_EzJOs0`.) Each car has a default velocity that it will maintain unless there is a car in the way. In that case, if there is room in an adjacent lane, the car will change lanes; otherwise, it will slow down. Gaussian noise is added to the observations of the cars' positions to emulate the inherent noise in physical sensors.

In addition to left and right lane change modes, we experimented with several possible sets of modes for the traffic data, including modes defined in terms of y velocity, headway, and a combination of y velocity and headway. The results presented here use left and right lane change modes and two different "stay-in-lane" modes for cars at different speeds (i.e., fast and slow).

We compare the accuracy of the learned MMPSR with two other prediction methods: a baseline method that predicts that the car will maintain its last observed velocity, and locally weighted linear regression trained upon the entire data set. We evaluated the models' predictions about how far each car would travel in the y direction over the future 2 and 5 seconds. The MMPSR performed significantly better than both comparison methods at both the 2 and 5-second horizons (Figure 8). Even in the presence of noisy observations, the MMPSR achieves less than 2% error (relative to distance traveled), for both 2 and 5 seconds in the future.

Mode	LLC	RLC	Slow	Fast
Predicted prob. of true mode	0.50	0.29	0.79	0.62
Fraction of data set	0.03	0.01	0.41	0.55

Fig. 9. Predicting the current mode for simulated traffic. "LLC" and "RLC" are left and right lane changes.

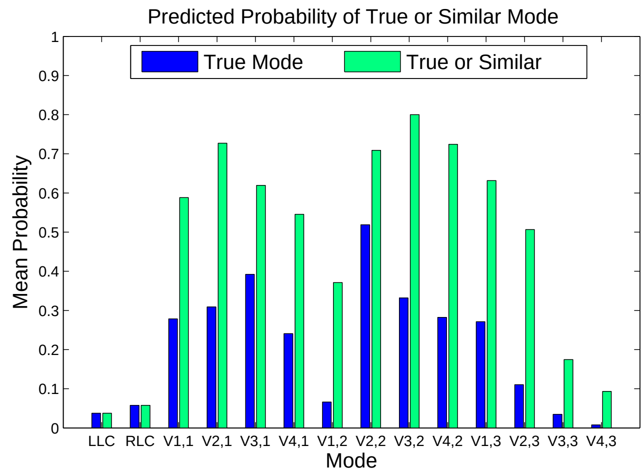

Fig. 10. Predicting the current mode for I-80 traffic: likelihood assigned to the true (or similar) modes by the learned MMPSR. Regarding the mode labels, LLC and RLC are left and right lane changes; the other labels "Vk,j" indicate the mode of a car that falls in the k^{th} velocity bin (4 is fastest) and j^{th} headway bin (3 is the longest distance to the car in front).

Not only does the MMPSR predict the movement of the cars accurately, it also assigns reasonably high likelihood to the true value of the current mode (Figure 9). The low likelihoods for the "lane change" modes are primarily due to the low prior probability of those modes (Figure 9).

2) Interstate 80 Traffic: We also learned an MMPSR to model real highway traffic on interstate 80 (I-80) near San Francisco. We used the data collected by the Next Generation SIMulation (NGSIM) project, which used cameras to capture traffic movement along approximately 500 meters of six-lane freeway [6]. (A video is available at `http://www.youtube.com/watch?v=JjxNu2kbtDI`.)

As with the simulated traffic, we experimented with several sets of mode definitions. The results presented here use modes determined by the y velocity and the headway to the car in front, which were discretized into four and three bins, respectively. Along with the two lane change modes, this gives fourteen total modes for the system. Partly because of the fine distinctions between the modes (cf. the simulated traffic with only two velocity bins), the MMPSR has more difficulty determining the true mode than with the previous domains (Figure 10). However, the likelihood that the MMPSR assigns to either the true mode or a similar mode (i.e., those that differ by one notch in velocity or headway, but not both) is significantly higher (Figure 10).

Even though the MMPSR does not always track the true mode, its predicted modes are close enough to enable good predictions about the cars' movements. These predictions were evaluated in the same way as for the simulated traffic,

2 sec.	L1 Error	Percent Error
MMPSR	4.185 ± 0.250	6.552 ± 0.228
Oracle MMPSR	3.615 ± 0.123	6.048 ± 0.258
Last Velocity	4.538 ± 0.161	6.601 ± 0.249
Local Regression	49.648 ± 2.416	99.258 ± 0.129

5 sec.	L1 Error	Percent Error
MMPSR	13.827 ± 0.501	9.092 ± 0.424
Oracle MMPSR	9.196 ± 0.263	6.265 ± 0.316
Last Velocity	17.216 ± 0.687	11.187 ± 0.511
Local Regression	110.548 ± 5.943	89.432 ± 0.547

Fig. 11. Error in predicting distance traveled for I-80 traffic. The "Oracle MMPSR" error provides a sense of the best error that could be achieved. Among the remaining models, the MMPSR achieves the lowest error for predicting both 2 and 5 seconds in the future. The confidence intervals are two standard deviations, computed across 15 data sets.

2 sec.	L1 Error	Percent Error
MMPSR	0.678 ± 0.120	99.057 ± 2.656
Local Regression	0.666 ± 0.027	102.652 ± 3.272
Last Velocity	0.908 ± 0.039	105.316 ± 1.918

5 sec.	L1 Error	Percent Error
MMPSR	1.348 ± 0.214	101.307 ± 4.148
Local Regression	1.301 ± 0.087	99.366 ± 0.352
Last Velocity	2.403 ± 0.103	135.781 ± 4.937

Fig. 12. Error in predicting lateral movement of I-80 traffic. The last-velocity comparison model has the highest error, while the MMPSR and local regression models are comparable. The confidence intervals in the tables are two standard deviations, computed across 15 data sets. The percent error is so high because the true lateral movement is sometimes small, so even small absolute errors can translate to large percent errors.

including a comparison with locally-weighted regression and predicting the last velocity. We also evaluated an MMPSR that uses an "oracle" feature that peers into the future. This feature is included in the low-level state, even though it will never be available online. Nevertheless, the oracle MMPSR's error provides a sense of the best error that could be achieved. Specifically, the oracle feature is *the desired prediction* (i.e., the distance the car will travel) minus the future value of the gap between it and the car it is following.[1]

As with the simulated traffic, the MMPSR performed better than both the local-regression and the last-velocity comparison methods at both the 2 and 5-second horizons (Figure 11), with a considerable difference at the 5-second horizon. Also, the error of the MMPSR is reasonably close to that of the oracle MMPSR, despite all the information contained in the oracle feature. Finally, it is worth noting that the MMPSR achieves less than ten percent error (relative to distance traveled), even when predicting five seconds in the future for cars at highway speeds. In absolute terms, this is about 14 feet, or roughly one car length.

Figure 12 shows the error in predicting *lateral* movement for the MMPSR, the last-velocity model, and locally-weighted linear regression. (The oracle MMPSR is not applicable to lateral movement.) The last-velocity model does the worst, while the MMPSR and local regression models are comparable. The percent error is so high because the true lateral movement is sometimes very small. For those time points, a small absolute error results in an enormous percent error, which skews the average percent error. However, in absolute terms, the average error of the MMPSR is quite small: less than 1.5 feet for five seconds in the future.

Neither the last-velocity model nor local regression performed well on both lateral and forward predictions. This is in contrast to the MMPSR, which is comparable to (or better than) the better of the comparison methods for both lateral and forward predictions.

III. Summary

We have introduced the MMPSR, a hierarchical model motivated by the problem of modeling vehicles' movements. In general, MMPSRs can model uncontrolled systems that switch between modes of behavior. Inspired by PSRs, the modes are not latent variables but are defined in terms of both historical and future observations. Because the modes are defined in terms of observations, learning the MMPSR model is more efficient than if the modes were latent variables. Furthermore, when using the MMPSR model to make predictions, the MMPSR can adjust its state to reflect the true values of the modes because those true values are eventually recognized from the observations. Even though the modes are defined in terms of observations, they are not restricted to be features of history, which would limit the expressiveness of the model. Rather, the mode definitions can include *future* observations, allowing modes such as a right lane change. We introduced a learning algorithm for the MMPSR, using it to learn MMPSR models of three systems, including simulated and real-world highway traffic. The MMPSR achieved lower error than comparison methods on all three systems, including the highway traffic on I-80.

Acknowledgements: The authors' (BW & SS) research was supported by NSF grant IIS-0413004. Any opinions, findings, conclusions or recommendations expressed here are those of the authors and do not necessarily reflect the views of the sponsors.

References

[1] W. S. Cleveland, S. J. Devlin, and E. Grosse. Regression by local fitting : Methods, properties, and computational algorithms. *Journal of Econometrics*, 37(1):87 – 114, 1988.

[2] S. Fine, Y. Singer, and N. Tishby. The hierarchical hidden markov model: Analysis and applications. In *Machine Learning*, pages 41–62, 1998.

[3] M. L. Littman, R. S. Sutton, and S. Singh. Predictive representations of state. In *Advances in Neural Information Processing Systems 14*, pages 1555–1561, 2002.

[4] E. J. Rafols, M. B. Ring, R. S. Sutton, and B. Tanner. Using predictive representations to improve generalization in reinforcement learning. In *Proceedings of the* 19^{th} *International Joint Conference on Artificial Intelligence*, pages 835–840, 2005.

[5] S. Singh, M. R. James, and M. Rudary. Predictive state representations: A new theory for modeling dynamical systems. In *Proceedings of the* 20^{th} *Conference on Uncertainty in Artificial Intelligence*, pages 512–519, 2004.

[6] U.S. Federal Highway Administration. Interstate 80 freeway dataset, 2006. http://www.tfhrc.gov/about/06137.htm.

[7] B. Wolfe, M. R. James, and S. Singh. Learning predictive state representations in dynamical systems without reset. In *Proceedings of the* 22^{nd} *International Conference on Machine Learning*, pages 985–992, 2005.

[1] This evaluation only considers the time points where there is a car in front of the modeled car; otherwise, the oracle feature is not defined. In the I-80 data set, there is almost always a car in front, so the vast majority of the data is evaluated.