Learning to Recognize Time Series: Combining ARMA models with Memory-based Learning

Kan Deng Andrew W. Moore Michael C. Nechyba

The Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213
kdeng@ri.cmu.edu awm@ri.cmu.edu nechyba@ri.cmu.edu

Abstract

For a given time series observation sequence, we can estimate the parameters of the AutoRegression Moving Average (ARMA) model, thereby representing a potentially long time series by a limited dimensional vector. In many applications, these parameter vectors will be separable into different groups, due to the different underlying mechanisms that generate differing time series. We can then use classification algorithms to predict the class of a new, uncategorized time series. For the purposes of a highly autonomous system, our approach to this classification uses memory-based learning and intensive cross-validation for feature and kernel selection. In an example application, we distinguish between driving data of a skilled, sober driver vs. a drunk driver, by calculating the ARMA model for the respective time series. In this paper, we first give a brief introduction to the theory of time series. We then discuss in detail our approach to time series recognition, using the ARMA model, and finish with experimental results.

1. Introduction

A time series is a sequence of signals or observations $\boldsymbol{x}_t$, each one recorded at a specified time instant *t*. For example, if we record the volume and tone of a piece of music at each time unit, we get a time series. Notice that each observation $\boldsymbol{x}_t$ consists of two variables, volume and tone. We might wish to categorize the mood of the music from the time-series. Another example is the observation sequence of the distances from a car's center to the road's center. We might wish to categorize the driving style; perhaps distinguishing between normal, drowsy, and drunk. Music and car position are two examples amongst a wide range of potentially useful time series categorization applications.

A *discrete time series* is one in which the signals or observations $\boldsymbol{x}_t$ are recorded at discrete time points. Without special notation, a discrete time series often refers to an observation sequence with fixed time interval between two neighboring observations. Our research handles only such discrete time series.

A time series is *multivariate* if there is more than one variable involved in each observation and if these variables are cross-related. In the music example above, if the volume and tone are cross-related, the music signal sequence is a multivariate time series. In this initial study, we only consider univariate time series.

We divide the time series recognition approaches into two classes: non-parametric and parametric ones. Parametric methods assume there is a model underlying the generation of the time series. The recognition of a time series is equivalent to the classification of the underlying models. HMM [Rabiner, 1989], [Hannaford & Lee, 1991] is such a model, [Nechyba & Xu, 1997] and [Pook & Ballard, 1993] used this model to do the time series recognitions in different domains. Other popular parametric models for time series are neural networks and recurrent networks [Elman, 1990], [Feldkamp, 1994], [Seawell & Kalman, 1995], [Nikovski, 1995].

Non-parametric methods do not use models. Instead, given some time series samples, they extract the features of these samples, such as the mean values, variances, correlations and frequencies, and do the recognition job based on these features. Non-parametric recognition researches include [Vandewalle & Moor, 1988], [Dellaert, Polzin & Waibel, 1996] and many more.

In this paper, we explore the parametric approach using a well-established model called AutoRegression Moving Average, or briefly ARMA*(p,q)*. ARMA model has been tried previously in the field of signal processing [Basseville & Nikiforov, 1993], especially speech recognition [Makhoul, 1975]. Compared with HMM, neural networks and recurrent networks, ARMA*(p,q)* is specially good at modeling stationary and Gaussian-distributed time series. And since ARMA*(p,q)* is a linear model, it can require vastly less computation to estimate the coefficients of the model. For our purposes ARMA*(p,q)* is an ideal model; for a certain time series sample, there is one and only one ARMA*(p,q)* model with specified coefficients corresponding to it. This property makes the recognition work much easier. Section 2 will discuss the relationship between our methods with other parametric methods. Compared with non-parametric methods, our ARMA*(p,q)* model-based recognition methods offer some information, such as lag influence and interactions, which is not easily summarized by features. Since feature-based recognition is domain-dependent, we will talk about the relationship between ARMA-based recognition with feature classification while we describe the sober-drunk driving experiments in Section 4.

Given a time series observation sequence, we can estimate the parameters of the ARMA model which (approximately) generates this time series. Hence, we represent a time series sequence (which might be very long) by a limited dimensional vector consisting of the ARMA parameters. Suppose we collect many time series samples generated by a drivers in two different sobriety conditions, either drunk or normal. We will show that the parameter vectors of the driving performance time series fall into two groups.

In this paper, we first give a brief introduction to the theory of time series, especially ARMA*(p,q)*. Then, we discuss in details our memory-based approach to time series recognition using ARMA model, and the relationship between our method with other approaches. Finally, we show promising preliminary experimental results for a driving simulator, where we distinguish the driving performance of a sober and a drunken driver.

2. ARMA(p,q) model

The most important property of time series is that the current signal x_t influenced by the previous signals x_{t-1}, x_{t-2}, etc. A general time series model may be formed as,

$$x_t = f(x_{t-1}, x_{t-2}, \ldots, u_{t-1}, u_{t-2}, \ldots) + \xi_t \,,$$

where u_t is the control knob and ξ_t is the white noise at time t.

We have a special interest in linear models because they are tractable and useful for many cases, for many years. One linear model looks like,

$$x_t = \alpha_1 x_{t-1} + \ldots + \alpha_p x_{t-p} + \gamma_1 u_{t-1} + \ldots + \xi_t \,.$$

Let's omit the control items u_{t-i} temporarily. The remaining equation is called *AutoRegression model, or AR(p)*, in which p is the window size.

More careful study of this model raises one question. If the model includes the noise ξ_t disturbing signal x_t, why should we ignore the past noise items ξ_{t-1}, ξ_{t-2}, etc.? A more sophisticated model is thus:

$$\begin{aligned} x_t &= \alpha_1(x_{t-1} - \xi_{t-1}) + \alpha_2(x_{t-2} - \xi_{t-2}) + \ldots \\ &\quad \ldots + \alpha_p(x_{t-p} - \xi_{t-p}) + \xi_t \end{aligned}$$

To be even more general, the time series model can be formed as,

$$\begin{aligned} x_t &= \alpha_1 x_{t-1} + \alpha_2 x_{t-2} + \ldots + \alpha_p x_{t-p} + \\ &\quad + \beta_1 \xi_{t-1} + \beta_2 \xi_{t-2} + \ldots + \beta_q \xi_{t-q} + \xi_t \end{aligned} \tag{1}$$

This is called AutoRegression Moving Average model, or ARMA*(p,q)* model. The α's are the coefficients of the AutoRegression part, and β's are the coefficients of the Moving Average part. The AR model's window size is p, while MA's is q. The size of the whole model is $p + q$. For different application domain, p and q may be different.

ARMA*(p,q)* defined in equation (1) is not a panacea for all kinds of time series, it assumes the time series is *stationary* and *invertible*. There are many cases of the violation of the stationarity and invertibility restrictions. Some of them can be easily pre-eliminated, and ARMA model is then still useful. Trend and seasonality are two examples, referring to [Brockwell & Davis, 1991] in the chapters about ARIMA model. More advanced explorations about non-stationary time series are summarized in [Tong, 1990]. Most music time series are not stationary and invertible, and they are more complicated than trend and seasonality. Hence, we don't plan to apply our new approach to music recognition, though it is a very appealing topic. We started from distinguishing sober and drunk drivers.

3. Memory-based time series recognition

Memory-based time series recognition consists of following four phases.

1. Model selection and validation

Given a certain application domain, first of all we need to select an adequate model for it. Because we presume ARMA*(p,q)* is the model we prefer, we want to figure out what values of p and q are the best. To be more careful, we also want to reconfirm that the ARMA*(p,q)* is qualified for this domain. One way to decide which p and q are best for a certain domain consists of three steps, 1. selecting several typical time series samples from this domain; 2. for each possible combination of p and q (usually both p and q are no bigger than 10), calculating the value of AIC, which is a selection criteria proposed by [Akaike, 1976], based on the time series samples selected; 3. The best p and q should correspond to the minimum AIC value. [Choi, 1992]

To reconfirm the validation of ARMA*(p,q)* with specified p and q for a certain domain, we can use *Portmanteau* testing method, referring to [S-plus].

2. Collection of time series samples

We collect numerous time series samples from this domain. For each of them, we estimate the parameters $\underset{\sim}{\hat{\alpha}}$ and $\underset{\sim}{\hat{\beta}}$ involved in the ARMA*(p,q)* model (the values of p and q are fixed for all the samples from a certain domain). The vectors of $\underset{\sim}{\hat{\alpha}}$ and $\underset{\sim}{\hat{\beta}}$ are the underlying mechanism of this time series; or in other words, this time series sample is a realization of the ARMA*(p,q)* model specified by $\underset{\sim}{\hat{\alpha}}$ and $\underset{\sim}{\hat{\beta}}$. Therefore, it is reasonable to represent a time series by its corresponding ARMA model parameters $\underset{\sim}{\hat{\alpha}}$ and $\underset{\sim}{\hat{\beta}}$. The estimation of $\underset{\sim}{\hat{\alpha}}$ and $\underset{\sim}{\hat{\beta}}$ is often done by Maximum Likelihood method [Brockwell & Davis, 1991].

If several time series samples shares the same $\underset{\sim}{\hat{\alpha}}$ and $\underset{\sim}{\hat{\beta}}$, it means these samples are homogeneous. Suppose we have several car position time series samples, all the samples are generated by the same driver, the same car with the same road condition. These time series samples are homogeneous, and ideally their $\underset{\sim}{\hat{\alpha}}$'s and $\underset{\sim}{\hat{\beta}}$'s should be the same, or at least their $\underset{\sim}{\hat{\alpha}}$'s and $\underset{\sim}{\hat{\beta}}$'s

While we collect the time series samples, we evaluate the physical and/or psychological properties of each of them. For the car performance series, we can label each series as drunken or sober depending on the driver's sobriety condition. Or, if we want to be more accurate, we may measure the alcohol level in the driver's breath as the evaluation value. Let's denote the evaluation value of a time series as v. The evaluation value v can be either categorical or continuous depending on the application domain and the evaluation method.

3. Construction of the knowledge memory

For a specific application domain, following phase 1 we find an adequate ARMA*(p,q)*. After phase 2, we collect many time series samples and represent each of them (which can be very long) using limited dimensional vectors of ARMA*(p,q)* model parameter $\underset{\sim}{\hat{\alpha}}$ and $\underset{\sim}{\hat{\beta}}$. Also, each time series sample has an evaluation value v.

Recalling the car position time series example, suppose we have collected N such driving performance series samples, we can construct a knowledge memory containing N data points. Each data point stands for a driving performance series sample. It consists of two parts, the ARMA*(p,q)* parameter vectors $\underset{\sim}{\hat{\alpha}}_i$ and $\underset{\sim}{\hat{\beta}}_i$, and the evaluation value v_i, $i = 1, ..., N$.

4. Query recognition

A *query* time series is defined as a time series observation sequence whose evaluation value is unknown. The objective of query recognition is to approximate a query time series' evaluation value. Still for the driving example, suppose we are given a car performance time series generated by a driver in unknown sobriety, our task is to judge if the driver is in the normal sobriety condition. This task can be done in three steps,

a. Estimating the parameters of the query series, $\hat{\underset{\sim}{\alpha}}_q$ and $\hat{\underset{\sim}{\beta}}_q$.

b. Calculating the distance from the query series to every data point in the memory. Let's define the distance from the query to the k'th data point in the memory as,

$$\begin{aligned}\text{dist}^2(k) = &\sum_{i=1}^{p} u_i(\hat{\alpha}_i(k) - \hat{\alpha}_i(q))^2 + \\ &\sum_{j=1}^{q} w_j(\hat{\beta}_j(k) - \hat{\beta}_j(q))^2\end{aligned} \tag{3}$$

in which $k = 1, ..., N$.

Usually not all the parameters are equally significant for recognition. we can insert weights, u_i and w_j, into the distance definition and assign high weight values to those more significant parameters so that they have bigger influence on the distance. We can use cross-validation method to decide the values of these weights.

If *dist(k)* is small enough, we claim that the query series is similar to a time series observed previously, which is represented by the k'th data point in this memory. The reason is that their underlying mechanisms, or their corresponding ARMA parameters are similar.

c. Based on the distances from the query time series to each data point in the memory, we can find the nearest memory data point, whose underlying mechanism is most similar to the query's. Therefore, it is reasonable to guess the evaluation value of the query time series is close the evaluation of the nearest neighbor's in the memory.

However, since there exists noise in the estimation of ARMA*(p,q)* parameters, it is not wise to bet the evaluation of the query only on one data point. Besides, it is costly to enumerate the distances from the query to all the data points in the memory. Therefore, we need more advanced techniques, such as *k*-nearest neighbor, kernel regression [Atkeson, Moore & Schaal, 1997], locally weighted logistic regression [Deng & Moore, 1997], logistic regression-based classifier, etc., and more efficient memory retrieval mechanism.

4. Related models

In this section, we discuss two other methods for time series classification: (1) Hidden Markov Models (HMMs) and (2) recurrent networks. First, a Hidden Markov Model consists of a set of states interconnected through probabilistic state transitions. The states themselves cannot be observed directly; rather each state outputs some observable, based on some output probability distribution. *Discrete* HMMs output discrete symbols, while *continuous* HMMs output continuous values, based on a mixture of Gaussian probability distributions. In both cases, the overall output of an HMM is generally neither uniform nor Gaussian distributed, but instead is more complicated. Unlike HMMs, the distribution of an ARMA time series is, however, asymptotically Gaussian distributed.

Second, recurrent neural networks are similar to feedforward neural networks, but allow connections in all directions, including self-connections and backward connections. As such, a recurrent network is theoretically capable of modeling any time series (assuming no limit on the number of nodes in the network). Depending on the structure and weights of the recurrent network, the output of the network can be either discrete or continuous, and will generally be distributed non-Gaussian. If we were to substitute a recurrent network for the ARMA model in our recognition approach, we could, however, not follow our outlined procedure. For example, if two time series are homogeneous, the weights of the corresponding recurrent networks may or may not be similar. In fact, for the same time series observation sequence, there may well exist infinitely many equivalent recurrent networks with different weights. There currently exists no good method for judging the similarity between recurrent networks with different weights.

Concerning computational complexity, the cost of the ARMA estimation algorithm is $O(m^3T)$, where T is the length of the time series sample, $m = \max(p, q+1)$. The cost of training an HMM is $O(N^2T^2)$, where N is the number of states in the HMM, and T is the length of the time series sample. In general, HMMs require more data to train, so that the T in the HMM's computational cost is usually larger than in the ARMA model. Therefore, the overall computational cost of training HMMs is larger than the cost of estimating the ARMA parameters. Also, the cost of training a recurrent neural network tends to be much higher than for the ARMA model. To summarize, we expect that the ARMA method will be more tractable than either of the above.

5. Experiments

In this section, we use our proposed method to distinguish driving performance between a sober, alert driver and an intoxicated driver. Since it is clearly unsafe to ask someone to driver while drunk, we use a dynamic driving simulator (see Figure 1) instead [Nechyba, 1997]. The user has independent control over steering (horizontal mouse position), the brake (left mouse button) and the accelerator (right mouse button). For a particular driver, the simulator records the distance from the road median at 5Hz (the simulator itself runs at 50Hz).

Prior to collecting actual data, we allowed the operator (let's call him Larry) to get used to the simulation interface. Larry is asked to stick to the right as much as possible and keep speed around 50m/hour. We then collected multiple data sets from Larry --- first, while he was sober, and second while he was under the influence. To collect driving data under the influence of alcohol, we had Larry consume approximately 150mL of alcohol (in the form of brandy). Within a half hour, Larry begins to feel the effects of the alcohol. For an hour thereafter, we collect driving data from Larry as he becomes increasingly intoxicated. While we classify the time series data as either "sober" or "drunken," the so-

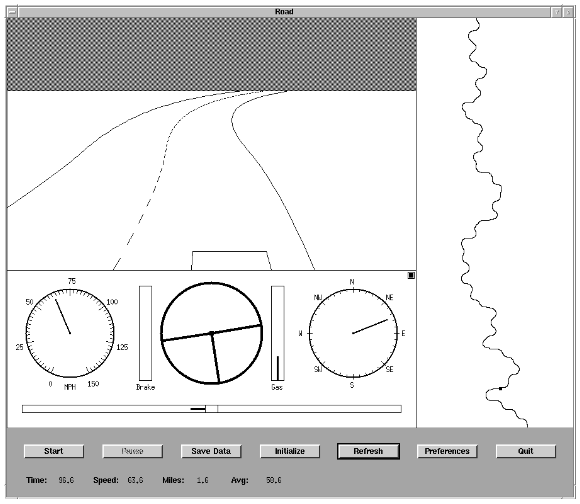

Fig. 1: Driving simulator interface (courtesy M. C. Nechyba).

briety levels of the “drunken” data clearly varies over time. Initially, Larry exhibits few side effects from the alcohol he had consumed 30 minutes before. Later, as he becomes increasingly intoxicated, however, he has significant difficulty staying on the road. Figure 2a shows an example of sober driving performance, while Figure 2b indicates an intoxicated driving performance (50 minutes after consuming the alcohol).

Once we have collected the time series observation sequences, we want to determine the appropriate model size p and q for the ARMA model. For two randomly chosen time series samples — one from the sober category, the other from the drunken category — we try different combinations of p and q, and estimate the corresponding α and β parameters. We then calculate the goodness-of-fit and AIC values for the corresponding ARMA models, as reported in Table 1 below. Clearly, the smallest AIC value corresponds to the ARMA(4,4) model. After selecting the appropriate model size, we estimate the ARMA(4,4) parameter vector for all N different time series samples.

Now, in order for our proposed method to work, we expect that the parameter vectors fall into two clusters, one corresponding to sober driving, the other corresponding to drunken driving. We also expect the “drunken” parameter vectors to be more widely scattered, due to the varying levels of intoxication experienced by Larry. Figure 4, which illustrates the clusters of parameter vectors for the driving time series, confirms our expectations.

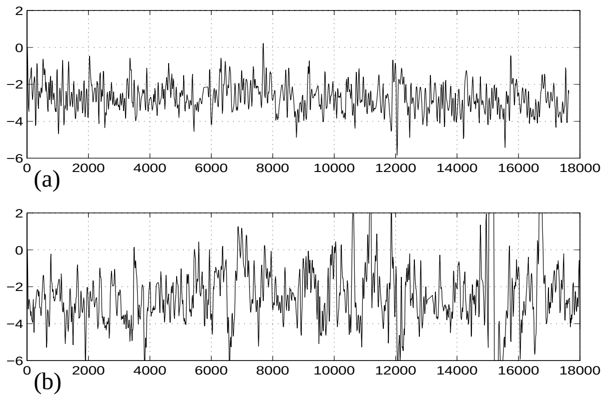

Fig. 2: Sober driving performance vs. drunken one.

Table 1: Selection an adequate ARMA(p,q) model

(p,q)	*(3, 3)*	*(4, 4)*	*(4, 5)*	*(5, 4)*	*(5, 5)*
Good-fit	0.2831	0.2772	0.2898	0.2827	0.2959
AIC	323.60	322.89	342.81	326.02	344.89

Looking at the results in Figure 4, two questions arise: (1) Why not just use variance to perform the classification task? and (2) Does the ARMA-based method give discrimination capability beyond that of simple variance? To answer the first question, we suggest that variance alone does not necessarily discriminate sufficiently between sober and intoxicated driving. For example, suppose a sober driver prefers to change lanes frequently while driving. In this case, the variance for his driving data would be quite large, and might be mistaken for intoxicated driving.

To answer the second question, we conduct the following experiment. First we calculate the standard deviations of the sober series (*0.8921*) and the drunken time series (*1.4775*). We then scale the drunken time series by *0.8921/1.4775 = 0.6038,* and again try our ARMA(4,4) estimation. If the ARMA model simply classifies based on variance, the algorithm should now fail. Figure 5 shows the distribution of ARMA parameter vectors for the sober and scaled drunken time series. Comparing Figures 4 and 5, we see that the ARMA parameters of the scaled drunken timer series are almost identical to those of the raw drunken time series. Thus, the ARMA model clearly offers information beyond simple variance. But why?

Recall the ARMA*(p,q)* model,

$$x_t = \alpha_1 x_{t-1} + \alpha_2 x_{t-2} + \dots + \alpha_p x_{t-p} + \\ + \beta_1 \xi_{t-1} + \beta_2 \xi_{t-2} + \dots + \beta_q \xi_{t-q} + \xi_t$$

When x_t is scaled by r, the α’s and β’s will ideally remain the same; the variance of the noise will, however, be scaled by r^2. Thus, our ARMA-based classification does not cue off variance. In fact, upon closer inspection, we conclude that response time, rather than variance, is the most critical difference between sober and drunken driving. Consider the α’s for the sober and drunken time series, respectively. For the sober time series, the magnitudes of α_2 and α_3 are most significant, indicating that the current state x_t is decided primarily by x_{t-2} and x_{t-3}. For the intoxicated time series, however, α_2 and α_3 are close to zero, such that x_{t-4} and car positions further removed in time primarily determine x_t. Thus, the response time of the drunk driver is significantly worse than that of the sober driver. To further confirm this observation, we will conduct simulated drunk driving experiments in the future, where we will blank the simulator interface from time to time in order to create an artificial lag in the response time of the driver. If, under these conditions, the driving performance of the sober drive is similar to the drunk driver, our conclusion would be correct.

Furthermore, note that the β values of the drunk ARMA model are scattered more widely than the sober β values. This implies that the drunk driver tends to jerk the driving controls significantly more than a sober driver. In the near future, we will test this observation by adding random noise to the driver controls, so as to further simulate drunk driving for a sober driver.

6. Discussion

In this preliminary experiment, we didn't involve in the control variables, such as the curvature of the road, traffic and speed, etc. To consider the influence of them, we can chop the driving time series to many pieces according the change of the control variables, and compare the ARMA parameters of those time series pieces with similar control variables.

Next, we plan to improve both the autonomy and computational efficiency of our algorithm. A robust classifier for time series data would be a useful tool and we hope that the combination of ARMA, AIC and intense cross validation of memory-based models will achieve that. We plan to use our time series recognition method to design a facility whose objective will be (1) to detect if a driver's sobriety condition is beyond acceptable bounds, and (2) to generate warning signals to the driver if necessary. The system will consist of three parts:

1. *Calibration module*, which measures the distance from the car's center to the lane's center,

2. *Time series module*, which does the estimation of the ARMA model and recognizes the time series based on the classification of the ARMA parameters.

3. *Signal module*, which gives the driver a proper signal, such as a blinking lamp, and/or a microphone.

[Pomerleau, 1995] has done some impressive work in autonomous driving. We can use their navigation system to build our calibration module. Our main contribution will be the time series module. Finally, the signal module can be done by engineers in Detroit. The overall system is illustrated in Figure 3.

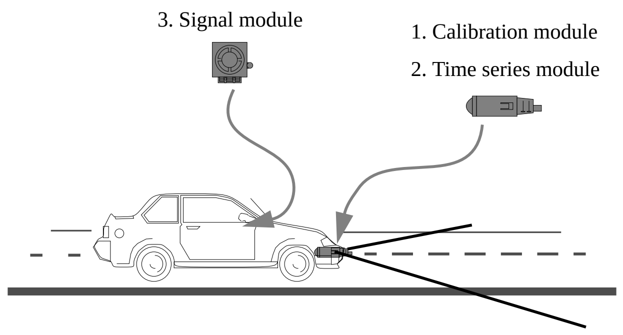

Fig. 3: Driving performance monitoring and warning system.

References

[Akaike, 1971] Akaike, H., *Autoregressive model fitting for control*, Ann. Inst. Statist. Math. 23. 1971.

[Akaike, 1976] Akaike, H., *Canonical correlation analysis of time series and the use of an information criterion*, in *System Identification: Advances and Case Studies*, R.K.Mehra and D.G.Lainiotis, Eds., Academic Press, New York, 1976.

[Atkeson, Moore & Schaal, 1997] Atkeson, C. G., Moore, A. W., and Schaal, S., *Locally Weighted Learning*, to appear in AI Review, 1997.

[Basseville & Nikiforov, 1993] Basseville, M., and Nikiforov, I. Detection of Abrupt Changes: Theory and Applications, 1993.

[Brockwell & Davis, 1991] Brockwell, P.J, and Davis, R.A., *Time Series: Theory and Methods*, second edition, 1991.

[Choi, 1992] Choi, B. S., *ARMA Model Identification*, by Springer-Verlag New York, Inc. 1992.

[Dellaert, Polzin & Waibel, 1996] Dellaert, F., Polzin, T., and Waibel, A., *Recognizing Emotion in Speech*. in ICSLP'96 Conference Proceedings, 1996.

[Dempster, Laird & Rubin, 1977] Dempster, A. P., Laird, N.M., and Rubin, D. B., *Maximum likelihood from incomplete data via the EM algorithm*. in *Journal of the Royal Statistical Society*, B 39 (1), 1977.

[Deng & Moore, 1997] Deng, K., and Moore, A.W., *Locally Weighted Logistics for Classification*, (in preparation).

[Elman, 1990] Elman, J.L, *Finding Structure in Time*, in Cognitive Science 14, pp. 179-211, 1990.

[Hannaford & Lee, 1991] Hannaford, B., and Lee, P., *Hidden Markov Model Analysis of Force/Torque Information in Telemanipulation*, in the International Journal of Robotics Research., Vol. 10, No. 5, October 1991.

[Kohn & Ansley, 1985] Kohn, R., and Ansley, C.F. *Estimation, prediction and interpolation for ARIMA models with missing data*, technical report, Graduate School of Business, University of Chicago. 1985.

[Makhoul, 1975] Makhoul, J. *Linear Prediction*, in *Proceedings of IEEE*, vol. 63, No. 4, April 1975.

[Nechyba & Xu, 1997] Nechyba, M.C., and Xu, Y., *Human Control Strategy: Abstraction, Verification and Replication*, to appear in *IEEE Control Systems Magazine*, April, 1997.

[Nikovski, 1995] Nikovski, D.N., *Adaptive Computation Techniques for Time Series Analysis*, A Master Degree Thesis, Dept. Computer Science, Southern Illinois University. July 1995.

[Pomerleau, 1995] Pomerleau, D., RALPH: *Rapidly Adapting Lateral Position Handler* in *1995 IEEE Symposium on Intelligent Vehicle*, Detroit, Michigan, U.S.A. 1995.

[Pook & Ballard, 1993] Pook, P. K., and Ballard, D.H., *Recognizing Teleoperated Manipulations*, in IEEE Control Systems Magazine, 1993.

[Puskorius & Feldkamp, 1994] Puskorius, G.V., and Feldkamp, L.A., *Neurocontrol of Nonlinear Dynamical Systems with Kalman Filter Trained Recurrent Networks*, in IEEE Transactions on Neural Networks, Vol. 5, No. 2, March 1994.

[Rabiner, 1989] Rabiner, L.R., *A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition*, in Proceedings of the IEEE, Vol. 77, No. 2, Feb. 1989.

[Seawell & Kalman, 1995] Seawell, T.I., and Kalman, B.L., *Time Variability While Training a Parallel Neural Net Network*, TR WUCS-95-27, Dept. Computer Science, Washington University, August 1995.

[S-plus] *Statistical Sciences, S-plus Guide to Statistical & Mathematical Analysis*, Version 3.3, Seattle: StatSci, a division of MathSoft, Inc. 1995

[Stoffer, 1986] Stoffer, D.S., *Estimation and Identification of Space-Time ARMAX Models in the Presence of Missing data.* Journal of the American Statistical Association, Sept. 1986, Vol. 81, No. 395, Theory and Methods.

[Tong, 1990] Tong, H., *Non-linear Time Series, A Dynamic System Approach*, published by Clarendon Press Oxford, 1990.

[Vandewalle & Moor, 1988] Vandewalle, J.,and De Moor B., *On the use of the singular value decomposition in identification and signal processing*. in the Proceeding of *the workshop of NATO Advanced Study Institute on Numerical Linear Algebra, Digital Signal Processing and Parallel Algorithms*, Leuven, Belgium, Aug. 1988.

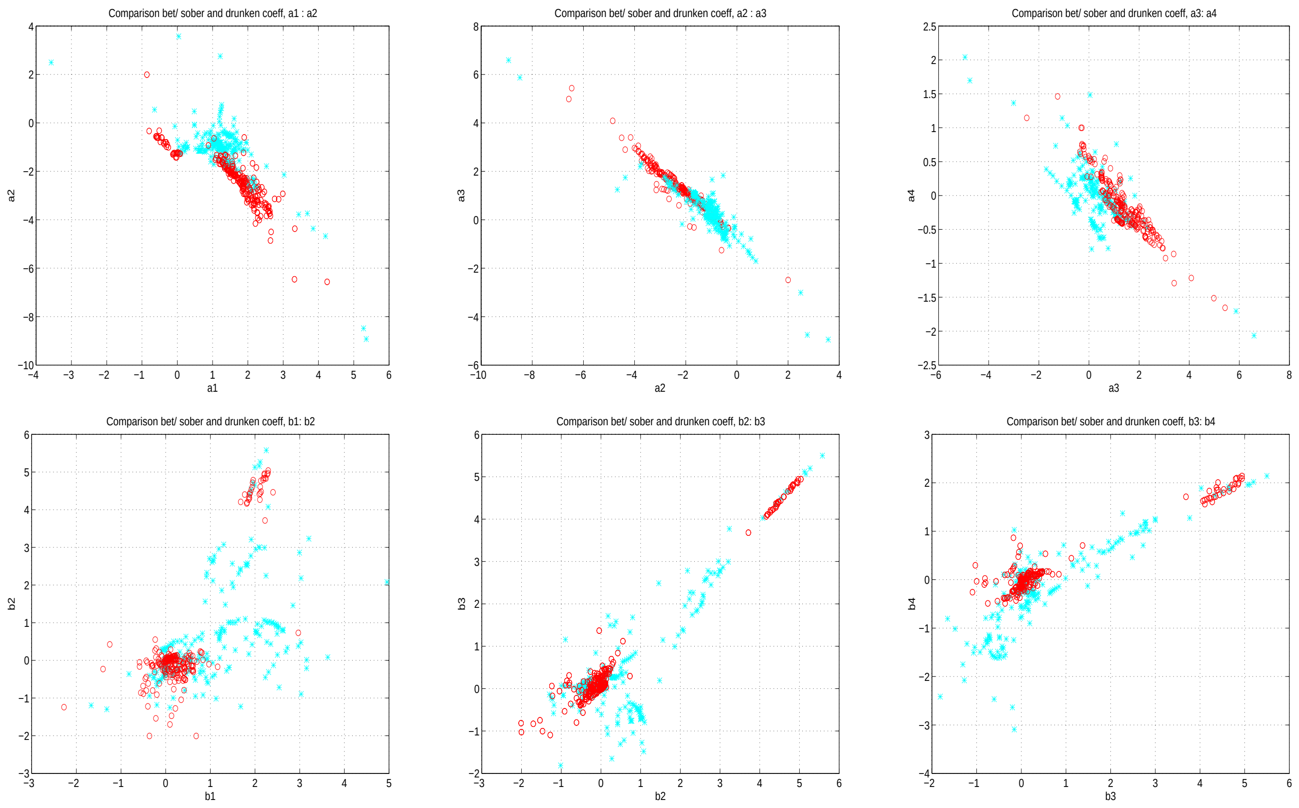

Fig. 4: ARMA(4,4) parameter clusters which distinguish driving performances from sober ones.

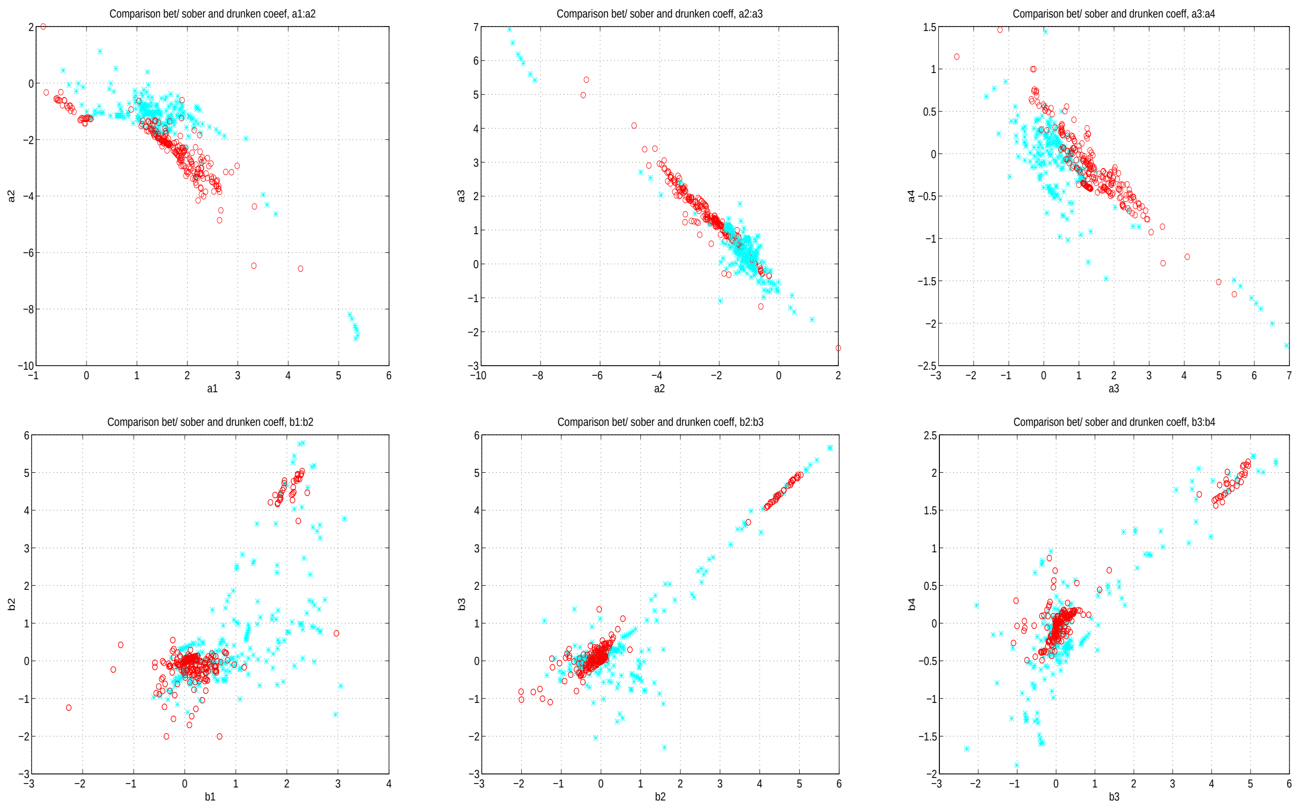

Fig. 5: The same as Figure 6, except that the drunken series have been scaled to have equal variances as the sober ones.