Bayesian Regression and Classification Using Mixtures of Gaussian Processes[*]

J.Q. Shi,
Depts. of Comp. Science & Statistics.
University of Glasgow,
Glasgow G12 8QQ,
Scotland, UK.
shi@dcs.gla.ac.uk

R. Murray-Smith,
Dept. of Computing Science,
University of Glasgow,
& Hamilton Institute,
National University of Ireland Maynooth.
rod@dcs.gla.ac.uk

D.M. Titterington,
Dept. of Statistics
University of Glasgow.
mike@stats.gla.ac.uk

Abstract

For a large data-set with groups of repeated measurements, a mixture model of Gaussian process priors is proposed for modelling the heterogeneity among the different replications. A hybrid Markov chain Monte Carlo (MCMC) algorithm is developed for the implementation of the model for regression and classification. The regression model and its implementation are illustrated by modelling observed Functional Electrical Stimulation experimental results. The classification model is illustrated on a synthetic example.

Keywords: Classification; Gaussian process; Heterogeneity; Hybrid Markov chain Monte Carlo; Mixture models; Nonlinear regression; Multiple models.

[*] Technical Report TR-2002-114, Department of Computing Science, University of Glasgow, Glasgow G12 8QQ, Scotland. June 2002

1 Introduction

Multiple model approaches to the empirical modelling of nonlinear systems have been of interest for many years, and have seen more widespread use in the last ten years. We reviewed the literature in (Johansen and Murray-Smith 1997), and recent years have seen a number of applications of the theory. However, subsequent work such as (Shorten *et al.* 1999, Leith and Leithead 1999) showed that there were problems with identification of parameters for off-equilibrium models, and that interpretation of local model parameters could often be misleading. This was generalised to the fuzzy modelling literature in (Johansen *et al.* 2000, Johansen and Babuska 2002). Sparseness of data in off-equilibrium regions of a nonlinear system's state-space often leads to high variance in location and parameters of off-equilibrium local models. A Markov chain Monte Carlo (MCMC) implementation of Bayesian multiple linear spline models which also provide suitable variance predictions is presented in (Holmes *et al.* 1999). In (Murray-Smith *et al.* 1999) we compared the multiple linear model approach with a nonparametric statistical model, the Gaussian process prior. Gaussian process priors gave high accuracy in model fit, combined with an accurate prediction of variance, which is especially important off equilibrium. Initially proposed by O'Hagan in (O'Hagan 1978), Gaussian process priors have recently been used in regression and classification (see recent reviews (Williams 1998, MacKay 1999, Williams and Barber 1998)). Mixtures of Gaussian processes have appeared in various forms (Shi *et al.* 2002, Rasmussen and Ghahramani 2002, Lemm 1999). However, a major disadvantage of the Gaussian process approach is that the implementation of the model requires the inversion of an $N \times N$ covariance matrix, for sample size N of training data, which has computational complexity of order $O(N^3)$, making the approach impractical for large data-sets.

A separate issue, but one often related to training set size, is how to deal with repeated groups or batches of measurements, where there is heterogeneity among the different replications (or groups). In many areas of empirical modelling we are faced with repeated experiments on similar objects and processes. In this paper, following on from (Shi *et al.* 2002), we propose a hierarchical model which deals with the issues of both grouped data and large training sets. A major advantage of this model and algorithm, is that it reduces the size of the covariance matrices being inverted, as they now correspond to the size of the training data for individual groups. As a consequence, the computational burden decreases dramatically. Inference for classification models based on Gaussian processes requires some extra development of the theory. Previous use of Gaussian processes in classification for single batches of data includes (Williams and Barber 1998), based on Laplacian approximations, and (Neal 1997), based on MCMC methods. This paper extends the hierarchical mixture approach to classification problems involving multiple batches.

2 The mixture model for regression

The model has a hierarchical structure: a lower-level model is applied separately to each group to model the basic structure of the data; the set of lower-level models have similar structures but with some mutual heterogeneity, i.e. informally, the groups are similar but slightly different, and a higher-level model is used among groups to model the heterogeneity. The Gaussian process prior model for regression or classification is used as the low-level model separately for each group. A mixture model, representing a good semi-parametric approach (see e.g. (Titterington *et al.* 1985)), is used for modelling the hierarchical structure. A hybrid MCMC algorithm is used for implementing inference.

2.1 Gaussian process priors

We are given N data points of training data $\{y_n, \boldsymbol{x}_n, n = 1, \cdots, N\}$, where $\boldsymbol{x}$ is a Q-dimensional vector of *inputs* (independent variables), and y is the *output* (dependent variable, target). A Gaussian process prior for regression is defined in such a way that $y(\boldsymbol{x})$ has a Gaussian prior distribution with zero mean and covariance function $C(\boldsymbol{x}_i, \boldsymbol{x}_j) = Cov(Y(\boldsymbol{x}_i), Y(\boldsymbol{x}_j))$. An example of such a covariance function is

$$C(\boldsymbol{x}_i, \boldsymbol{x}_j; \boldsymbol{\theta}) = v_0 \exp\left(-\frac{1}{2}\sum_{q=1}^{Q} w_q (x_{iq} - x_{jq})^2\right) + a_0 + a_1 \sum_{q=1}^{Q} x_{iq} x_{jq} + \delta_{ij}\sigma_v^2, \quad (1)$$

where $\boldsymbol{\theta} = (w_1, \cdots, w_Q, v_0, a_0, a_1, \sigma_v^2)$, and $\delta_{ij} = 1$ if $i = j$ and 0 otherwise. This covariance function is often used in practice. The first term recognises high correlation between the outputs of cases with nearby inputs, while the rest are a bias term, a linear regression term and a noise term respectively; see (O'Hagan 1978) and (Williams and Rasmussen 1996) among others. More discussion about the choice of covariance function can be found in (MacKay 1999).

Given a covariance function, the log-likelihood of the training data is

$$L(\boldsymbol{\theta}) = -\frac{1}{2}\log|\boldsymbol{\Psi}| - \frac{1}{2}\boldsymbol{y}^T \boldsymbol{\Psi}^{-1} \boldsymbol{y} - \frac{N}{2}\log 2\pi, \quad (2)$$

where $\boldsymbol{\Psi} = \boldsymbol{\Psi}(\boldsymbol{\theta})$ is the covariance matrix of $\boldsymbol{y} = (y_1, \cdots, y_N)^T$ with dimension $N \times N$. The maximum likelihood estimate (MLE) of $\boldsymbol{\theta}$ can be calculated by maximizing the above log-likelihood. An iterative optimization method, such as the conjugate gradient method, can be applied. It requires the evaluation of $\boldsymbol{\Psi}^{-1}$, which takes time $O(N^3)$. Efficient implementation with particular reference to approximation of the matrix inversion has been well developed; see for example (Gibbs 1997). However, it still becomes time-consuming and impractical for larger sets of training data (e.g. $N > 1000$).

If prior information is to be incorporated, a Bayesian approach is generally used.

Let $p(\boldsymbol{\theta})$ be the prior density function of $\boldsymbol{\theta}$ and let $\mathcal{D} = \{\boldsymbol{y}, \boldsymbol{x}\}$ be the training data. Then the posterior density of $\boldsymbol{\theta}$ given the training data is

$$p(\boldsymbol{\theta}|\mathcal{D}) \propto p(\boldsymbol{\theta})p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\theta}), \tag{3}$$

where $p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\theta})$ is the density function of an N-dimensional multivariate normal distribution with zero mean and covariance matrix $\boldsymbol{\Psi}(\boldsymbol{\theta})$, as defined by (1), for example. Since the form of the covariance function is complicated in terms of $\boldsymbol{\theta}$, it is infeasible to do any analytical inference based on the above posterior distribution. A Markov chain Monte Carlo approach is generally used; see (Neal 1997, MacKay 1999).

One major goal in engineering and other fields is to predict an output based on the training data. This problem can be solved thanks to the nice analytical properties of Gaussian processes. Let $\boldsymbol{x}^*$ be the test inputs. The predictive distribution can be obtained by conditioning on the observed outputs of the N training cases. Since the joint distribution for the outputs of the training cases and test cases is Gaussian, the predictive distribution is also Gaussian. Let $\boldsymbol{\Psi}^*$ be the covariance matrix of $(y_1, \cdots, y_N, y^*)$, where y^* denotes the output given $\boldsymbol{x}^*$. It is partitioned as

$$\boldsymbol{\Psi}^* = \left[\begin{array}{cc} \boldsymbol{\Psi} & \boldsymbol{\psi}(\boldsymbol{x}^*) \\ \boldsymbol{\psi}^T(\boldsymbol{x}^*) & C(\boldsymbol{x}^*, \boldsymbol{x}^*) \end{array} \right],$$

where $\boldsymbol{\psi}(\boldsymbol{x}^*) = (C(\boldsymbol{x}^*, \boldsymbol{x}_1), \cdots, C(\boldsymbol{x}^*, \boldsymbol{x}_N))^T$. The predictive distribution of y^* is therefore a Gaussian distribution with mean and variance given by

$$\begin{aligned} \hat{y}^* &= \boldsymbol{\psi}^T(\boldsymbol{x}^*)\boldsymbol{\Psi}^{-1}\boldsymbol{y}, & (4) \\ \hat{\sigma}^{*2} &= C(\boldsymbol{x}^*, \boldsymbol{x}^*) - \boldsymbol{\psi}^T(\boldsymbol{x}^*)\boldsymbol{\Psi}^{-1}\boldsymbol{\psi}(\boldsymbol{x}^*). & (5) \end{aligned}$$

The mean (4), evaluated at the MLE of $\boldsymbol{\theta}$, is generally used as a prediction of y^*. An alternative way of predicting y^* is to investigate its Bayesian predictive density, given by $p(y^*|\mathcal{D}, \boldsymbol{x}^*) = \int p(y^*|\mathcal{D}, \boldsymbol{x}^*, \boldsymbol{\theta})p(\boldsymbol{\theta}|\mathcal{D})d\boldsymbol{\theta}$. The integral can be approximated by using a Markov chain to sample the posterior density $p(\boldsymbol{\theta}|\mathcal{D})$. Suppose a set of samples $\{\boldsymbol{\theta}^{(1)}, \cdots, \boldsymbol{\theta}^{(S)}\}$ is generated from $p(\boldsymbol{\theta}|\mathcal{D})$. Then the predictive density is approximated by $p(y^*|\mathcal{D}, \boldsymbol{x}^*) \simeq \frac{1}{S}\sum_{s=1}^{S} p(y^*|\mathcal{D}, \boldsymbol{x}^*, \boldsymbol{\theta}^{(s)})$. The predictive mean is therefore $\sum_{s=1}^{S} E(y^*|\mathcal{D}, \boldsymbol{x}^*, \boldsymbol{\theta}^{(s)})/S$, where $E(y^*|\mathcal{D}, \boldsymbol{x}^*, \boldsymbol{\theta})$ is given by (4) for the particular value $\boldsymbol{\theta}$. The above quantity can be used as a prediction.

2.2 Hierarchical mixture models

The lower-level basic models described in the previous section are defined to fit the data corresponding to each replication (within a group) separately. The structures of the basic models are similar but with some mutual heterogeneity; a higher-level model is defined to model the heterogeneity among different replications (groups). In this section, we define a mixture regression model of Gaussian processes.

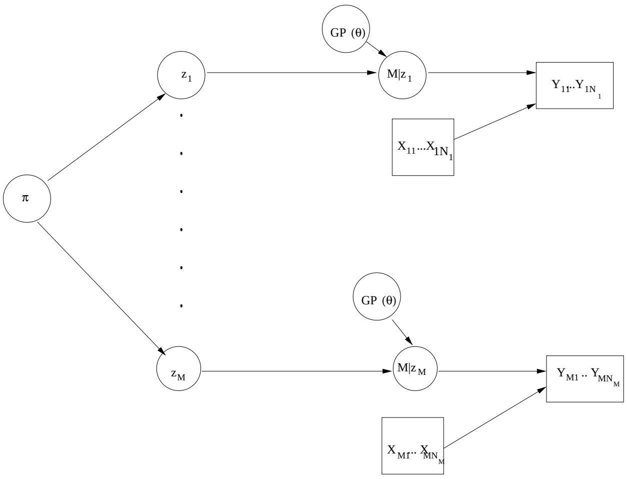


Figure 1: Hierarchical structure of model showing how M indicator variables select from multiple sub-models (each a Gaussian process) conditioned on M subsets of training data. Rectangles indicate observed values.

Suppose that there are M different groups of data (replications). In the mth group, N_m observations are collected. Let the observations be $y_{mn},\ m = 1, \cdots, M,\ n = 1, \cdots, N_m$. In a hierarchical mixture model of Gaussian processes for regression we have that

$$y_{mn}|z_m = k \ \sim \ GP_k(\boldsymbol{\theta}), \tag{6}$$

where z_m is an unobservable latent indicator variable. If $z_m = k$ is given, the model for group m is a Gaussian process regression model $GP_k(\boldsymbol{\theta})$. A special case which we use here corresponds to $GP_k(\boldsymbol{\theta}) = GP(\boldsymbol{\theta}_k)$, i.e., for different $GP_k(\boldsymbol{\theta})$, they have exactly the same structure and covariance function, but with different values of the parameter $\boldsymbol{\theta}_k$. The association among the different groups is introduced by the latent variable z_m, for which

$$P(z_m = k) = \pi_k, \quad k = 1, \cdots, K, \tag{7}$$

for each m. K is the number of components of the mixture model. We assume that K has a fixed given value in this paper.

We adopt the Bayesian approach. Let $\boldsymbol{\Theta} = (\boldsymbol{\theta}_1, \cdots, \boldsymbol{\theta}_K)$ and $\boldsymbol{\pi} = (\pi_1, \cdots, \pi_K)$, and let $\mathcal{D}$ be the collection of training data. The posterior marginal density of the unknown parameters is given by

$$p(\boldsymbol{\Theta}, \boldsymbol{\pi}|\mathcal{D}) \propto p(\boldsymbol{\Theta}, \boldsymbol{\pi})p(\mathcal{D}|\boldsymbol{\Theta}, \boldsymbol{\pi}), \tag{8}$$

where $p(\mathcal{D}|\boldsymbol{\Theta}, \boldsymbol{\pi}) = \prod_{m=1}^{M} \sum_{k=1}^{K} \pi_k p(\boldsymbol{y}_m|\boldsymbol{\theta}_k, \boldsymbol{x}_m)$. We assume that, *a priori*, $\boldsymbol{\Theta}$ and $\boldsymbol{\pi}$ are independent, and the $\boldsymbol{\theta}_k$ are independent and identically distributed, so that $p(\boldsymbol{\Theta}, \boldsymbol{\pi}) = p(\boldsymbol{\pi}) \ \prod_{k=1}^{K} p(\boldsymbol{\theta}_k)$. We will use the covariance function defined in (1), and adopt the priors given in (Rasmussen 1996); see also (Neal 1997)). As in the general setting of mixture models, we assume that $(\pi_1, \cdots, \pi_K)$ has a Dirichlet distribution, i.e. $p(\pi_1, \cdots, \pi_k) \sim D(\delta, \cdots, \delta)$, with $\delta = 1$, for example.

Obviously, it is almost impossible to do analytical posterior analysis based on the marginal density (8), so we use a hybrid MCMC algorithm. The main idea is to generate a sample of $(\boldsymbol{\Theta}, \boldsymbol{\pi}) = \{\boldsymbol{\theta}_k, \pi_k, k = 1, \cdots, K\}$ from its marginal posterior density (8). From our study, we found that the implementation is much more simple and efficient if the latent variable $\boldsymbol{z} = (z_1, \cdots, z_M)$ is augmented along with the unknown parameter $\boldsymbol{\Theta}$ of interest. Each sweep of this procedure, based on the Gibbs sampler, involves the following steps: (a) update $\boldsymbol{z}$ from $p(\boldsymbol{z}|\boldsymbol{\Theta}, \mathcal{D})$ given the current value of $\boldsymbol{\Theta}$; and (b) update $\boldsymbol{\Theta}$ from $p(\boldsymbol{\Theta}|\boldsymbol{z}, \mathcal{D})$ given the current value of $\boldsymbol{z}$. The details are given in (Shi *et al.* 2002).

2.3 Posterior analysis and prediction

Using the algorithm discussed in the last subsection, we generate samples of the parameter of interest $\boldsymbol{\Theta}$ and the latent indicator variable $\boldsymbol{z}$ from their posterior distribution. Denote the set of samples by $\{\boldsymbol{\theta}_1^{(s)}, \cdots, \boldsymbol{\theta}_K^{(s)}, \boldsymbol{z}^{(s)}, s = 1, \cdots, S\}$. From this set of samples, we approximate the predictive distribution for the output from a test input $\boldsymbol{x}^*$ in the mth group by

$$\begin{aligned} p(\boldsymbol{y}^*|\mathcal{D}, \boldsymbol{x}^*) &= \int p(\boldsymbol{y}^*|\mathcal{D}, \boldsymbol{x}^*, \boldsymbol{\theta}, z_m) p(\boldsymbol{\theta}, z_m|\mathcal{D}) d\boldsymbol{\theta} dz_m \\ &\simeq \frac{1}{S} \sum_{s=1}^{S} p(\boldsymbol{y}^*|\mathcal{D}, \boldsymbol{x}^*, \boldsymbol{\theta}^{(s)}, z_m^{(s)}). \end{aligned} \tag{9}$$

The predictive distribution $p(\boldsymbol{y}^*|\mathcal{D}, \boldsymbol{x}^*, \boldsymbol{\theta}^{(s)}, z_m^{(s)})$ is Gaussian with mean (4) and variance (5). In general, we use as a prediction the predictive mean, which is

$$\hat{y}_m^* = (\hat{y}_m^{*(1)} + \cdots + \hat{y}_m^{*(S)})/S, \tag{10}$$

where $\hat{y}_m^{*(s)}$ is given by (4) for the particular value $\boldsymbol{\theta}^{(s)}$. The variance associated with the prediction can be calculated similarly:

$$\hat{\sigma}_m^{*2} = \sum_{s=1}^{S} \hat{\sigma}_m^{*2(s)}/S + \sum_{s=1}^{S} (\hat{y}_m^{*(s)})^2/S - (\hat{y}_m^*)^2, \tag{11}$$

where $\hat{\sigma}_m^{*2(s)}$ is given by (5).

If we do not know to which particular group the test input $\boldsymbol{x}^*$ belongs, we may suppose that this test point is in the mth group with probability M^{-1} for all $m = 1, \cdots, M$. Therefore, the prediction is

$$\hat{y}^* = \sum_{m=1}^{M} \hat{y}_m^*/M \tag{12}$$

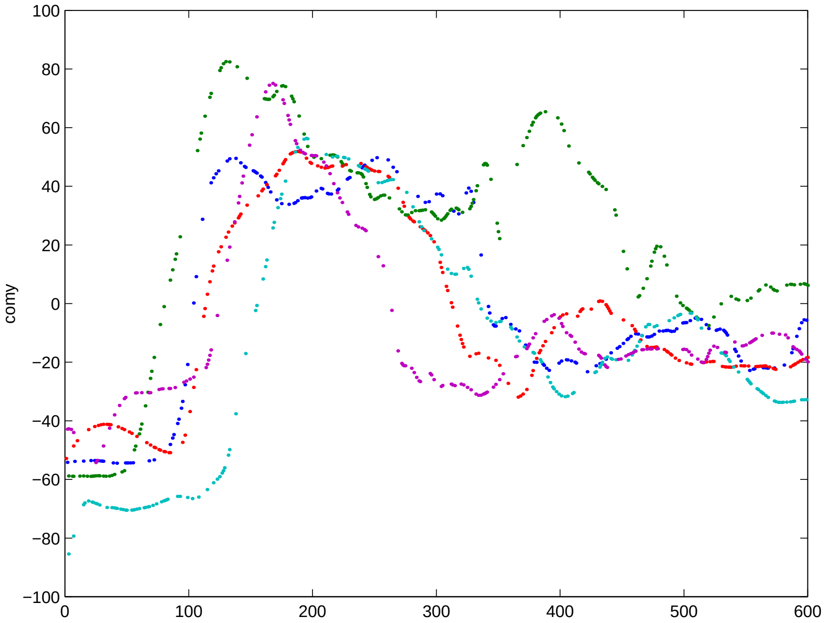
and the variance is

$$\hat{\sigma}^{*2} = \sum_{m=1}^{M} \hat{\sigma}_m^{*2}/M + \sum_{m=1}^{M} \hat{y}_m^{*2}/M - \hat{y}^{*2}, \tag{13}$$

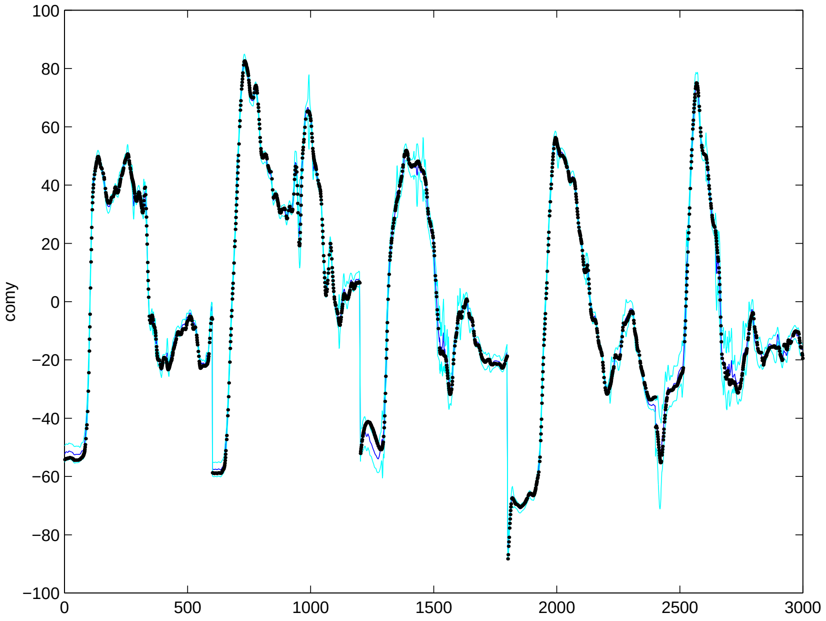
where $\hat{y}_m^*$ and $\hat{\sigma}_m^{*2}$ are given by (10) and (11) respectively. Note that $\hat{\sigma}^{*2}$ is larger than the average of the variances, $\sum_{m=1}^{M} \hat{\sigma}_m^{*2}/M$. More formally, a so called *allocation model* can be used to model the indicator z by the information about each group, such as the height, weight, age, the level of injury, the particular technique used in standing-up and so on of a patient in the FES example discussed in the next subsection.

2.4 Illustrative regression example

To illustrate this approach we provide a data-set of 5 standing-up trajectories for a single paraplegic patient stimulated by Functional Electric Stimulation (Kamnik and Bajd 1999, Shi *et al.* 2002). Six hundred data points are recorded for each standing-up. The trajectories of the body centre of mass (COM) are presented in Figure 2(a), which shows that the basic model structure for the five trajectories should be the same, while heterogeneity is also obvious. Thus, our hierarchical mixture model of Gaussian processes seems well suited to this problem. From the whole data-set of 3000 data points, we randomly select one third of the data points as training data; the rest are used as test data. The sample sizes of training data for the five groups are 186, 212, 211, 196 and 195 respectively. We use the hierarchical mixture model (6) and (7) with two components. The MCMC algorithm converges quickly (after about 600 iterations), and 80 samples are taken to predict the test data using equation (10). The close fit of the predictions to the measured data is shown in Figure 2(b). The root mean squared error is rmse=1.6020, while the sample correlation coefficient is $r = 0.9992$. For comparison, the results for a single Gaussian process regression model with the same covariance function (1) were poorer, with rmse=2.7123 and $r = 0.9973$.



(a) COM for 5 trajectories for one patient. Note that although the runs are qualitatively similar, there is heterogeneity among groups. This would be even more pronounced if multiple patients were included.



(b) Prediction results, with 2σ intervals, and observed data.

Figure 2: Data and model predictions for a single patient

3 Mixture models for classification

3.1 Latent Gaussian process model

Models for classification problems can be defined in term of a Gaussian process model for latent variables that are associated with each case. For binary classification, the target is from the set $\{0, 1\}$, and the input may be a vector of covariates $\boldsymbol{x}$. If we assume that the observation is $\{t_i, \boldsymbol{x}_i, i = 1, \cdots, n\}$, a logistic model can be defined in terms of continuous latent variable y_i as follows:

$$P(t_i = 1|y_i) = \frac{1}{1 + \exp(-y_i)}. \tag{14}$$

Another important model is the following probit model:

$$t_i = \begin{cases} 0 & \text{if } y_i < 0; \\ 1 & \text{if } y_i \geq 0. \end{cases} \tag{15}$$

y_i is also a continuous latent variable. $P(t_i = 1) = P(y_i \geq 0)$. We will use the logistic model in this paper.

The latent values y_i are given a Gaussian process prior:

$$\boldsymbol{y} = (y_1, \cdots, y_n)' \sim N(\mathbf{0}, \boldsymbol{C}). \tag{16}$$

A possible covariance function is (Neal 1997)

$$\boldsymbol{C}(\boldsymbol{x}_i, \boldsymbol{x}_j; \boldsymbol{\theta}) = v_0 \exp\left(-\frac{1}{2}\sum_{q=1}^{Q} w_q (x_{iq} - x_{jq})^2\right) + \delta_{ij} J^2, \tag{17}$$

where J defines the amount of 'jitter', which is similar to the noise in a regression model, included for improving the efficiency of sampling.

For multiple classification problems, the targets are from the set $\{0, 1, \cdots, V-1\}$. An analogous model can be defined using V latent variables $y_{i,v},\ v = 0, \cdots, V-1$ as follows

$$P(t_i = v|y_{i,0}, \cdots, y_{i,V-1}) = \frac{\exp(-y_{i,v})}{\sum_{u=0}^{V-1} \exp(-y_{i,u})}. \tag{18}$$

Here, the $y_{i,v}$ are assigned V independent Gaussian process priors as in (16).

3.2 Hierarchical mixture models for classication

Using a similar idea to that described in section 2.2, we propose to use a hierarchical mixture model to fit a large data set with repeated measurements for classification problems. Assume the data are obtained from M groups. Let the observations be $\{(t_{mi}, \boldsymbol{x}_{mi}), m = 1, \cdots, M, i = 1, \cdots, N_m\}$. V latent variables $y_{mi,v}$ can be defined by (18). We apply a latent Gaussian process model as a lower-level model separately to each group:

$$\boldsymbol{y}_{m,v} = (y_{m1,v}, \cdots, t_{mN_m,v})'|z_m = k \sim GP(\boldsymbol{\theta}_k) \tag{19}$$

independently for $v = 0, 1, \cdots, V-1$, where $GP(\boldsymbol{\theta}_k)$ is a Gaussian process model as in (16) with unknown parameter $\boldsymbol{\theta}_k$. The heterogeneity among the different groups is modelled by a mixture model as in equation (7), and we assume that the associated $(\pi_1, \cdots, \pi_K)$ has a Dirichlet prior distribution $D(\delta, \cdots, \delta)$.

3.2.1 Implementation of MCMC

The above model is analysed by the Bayesian approach, implemented by an MCMC algorithm. The basic idea is to generate a set of samples of the unknown parameter $\boldsymbol{\Theta} = \{\boldsymbol{\theta}_{k,v}, k = 1, \cdots, K\}$, and the latent variables $\boldsymbol{z} = (z_1, \cdots, z_M)$ and $\boldsymbol{y}$, from their posterior distributions. One sweep of the MCMC algorithm includes the following steps:

(a) update $\boldsymbol{z}$ from $p(\boldsymbol{z}|\boldsymbol{y}, \boldsymbol{\Theta}, \mathcal{D})$;

(b) update $\boldsymbol{\Theta}$ from $p(\boldsymbol{\Theta}|\boldsymbol{y}, \boldsymbol{z}, \mathcal{D})$;

(c) update $\boldsymbol{y}$ from $p(\boldsymbol{y}|\boldsymbol{\Theta}, \boldsymbol{z}, \mathcal{D})$.

Here, $\mathcal{D} = (\boldsymbol{t}, \boldsymbol{x})$ is the training data. The details of the subalgorithms are as follows.

(a) Updating $\boldsymbol{z}$ from $p(\boldsymbol{z}|\boldsymbol{y}, \boldsymbol{\Theta}, \mathcal{D})$

Let c_k be the number of observations for which $z_m = k$, over all $m = 1, \cdots, M$. We use a sub-Gibbs sampler in this step by introducing $\boldsymbol{\pi} = (\pi_1, \cdots, \pi_K)$. Similarly to the discussion in (Shi *et al.* 2002), one sweep of the procedure for sampling $\boldsymbol{z}$ and $\boldsymbol{\pi}$ is as follows:

(i) sample z_m from $p(z_m = k|\boldsymbol{y}, \boldsymbol{\Theta}, \boldsymbol{\pi}) \propto \pi_k \prod_{v=0}^{V-1} p(\boldsymbol{y}_{m,v}|\boldsymbol{\theta}_k)$;

(ii) sample $(\pi_1, \cdots, \pi_K)$ from $p(\pi_1, \cdots, \pi_K) \sim D(\delta + c_1, \cdots, \delta + c_K)$.

In this approach, a sample of $\boldsymbol{\pi}$ is also generated.

(b) Updating Θ from $p(\Theta|\boldsymbol{y},\boldsymbol{z},\mathcal{D})$

If the prior distributions of $\boldsymbol{\theta}_k$ are independent for different k, the conditional density function of Θ is

$$p(\Theta|\boldsymbol{y},\boldsymbol{z},\mathcal{D}) = \prod_{k=1}^{K} p(\boldsymbol{\theta}_k|\boldsymbol{y},\boldsymbol{z},\mathcal{D})$$

with

$$p(\boldsymbol{\theta}_k|\boldsymbol{y},\boldsymbol{z},\mathcal{D}) \propto p(\boldsymbol{\theta}_k) \prod_{m\in\{z_m=k\}} \prod_{v=0}^{V-1} p(\boldsymbol{y}_{m,v}|\boldsymbol{\theta}_k).$$

Thus the $\boldsymbol{\theta}_k$'s are conditionally independent, and we can deal with each of them separately. For a particular k, a hybrid MCMC algorithm similar to the one discussed in (Shi *et al.* 2002) can be used.

(c) Updating $\boldsymbol{y}$ from $p(\boldsymbol{y}|\Theta,\boldsymbol{z},\mathcal{D})$

Let $\boldsymbol{y}_m$ be the collection of $(y_{m1,v},\cdots,y_{mn_m,v})$ for all $v=0,\cdots,V-1$, then

$$p(\boldsymbol{y}|\Theta,\boldsymbol{z},\mathcal{D}) = \prod_{m=1}^{M} p(\boldsymbol{y}_m|\boldsymbol{\theta}_{z_m},\mathcal{D}) \propto \prod_{m=1}^{M} p(\boldsymbol{y}_m|\boldsymbol{\theta}_{z_m})p(\boldsymbol{t}_m|\boldsymbol{y}_m). \tag{20}$$

The conditional distributions of the $\boldsymbol{y}_m$ are therefore independent for $m=1,\cdots,M$. Within group m,

$$\begin{aligned} p(\boldsymbol{y}_m|\Theta,\boldsymbol{z},\mathcal{D}) &\propto p(\boldsymbol{y}_m|\boldsymbol{\theta}_{z_m})p(\boldsymbol{t}_m|\boldsymbol{y}_m) \\ &\propto \prod_{v=0}^{V-1} p(\boldsymbol{y}_{m,v}|\boldsymbol{\theta}_{z_m}) \prod_{i=1}^{n_m} \frac{\exp(y_{mi,t_{mi}})}{\sum_{v=0}^{V-1}\exp(y_{mi,v})}. \end{aligned}$$

This is log-concave for $y_{mi,v}$ (i.e., the log-density is concave). A Gibbs sampler with adaptive rejection sampling (Gilks and Wild 1992) is a very efficient way of sampling $\boldsymbol{y}_m$ from the above log-concave density function.

3.2.2 Prediction

The predictive probability at a new point $\boldsymbol{x}^*$ can be expressed as

$$P(t^*=1|\boldsymbol{x}^*,\mathcal{D}) = \int P(t^*=1|y^*)p(y^*|\boldsymbol{x}^*,\mathcal{D})dy^*. \tag{21}$$

A Monte Carlo method is used to generate a set of samples $\{y^{*(s)}, s=1,\cdots,S\}$ from $p(y^*|\boldsymbol{x}^*,\mathcal{D})$, and to approximate the integral required for calculating $P(t^*=1|\boldsymbol{x}^*,\mathcal{D})$ by

$$\sum_{s=1}^{S} P(t^*=1|y^{*(s)})/S, \tag{22}$$

where $P(t^* = 1|y^{*(s)})$ is given by (14).

The predictive distribution of the latent variable $p(y^*|\boldsymbol{x}^*, \mathcal{D})$ can be approximated by (9). Therefore, if we know that the test point belongs to the mth group, the above predictive distribution is approximately a Gaussian distribution with mean (10) and variance (11); otherwise, the mean and variance are given by (12) and (13) respectively.

Applying the model in (18) and the above procedure to $V - 1$ independent latent variables $y_{i,v},\ v = 0, \cdots, V-1$, we can deal with multiple classification problems.

3.3 Classification example

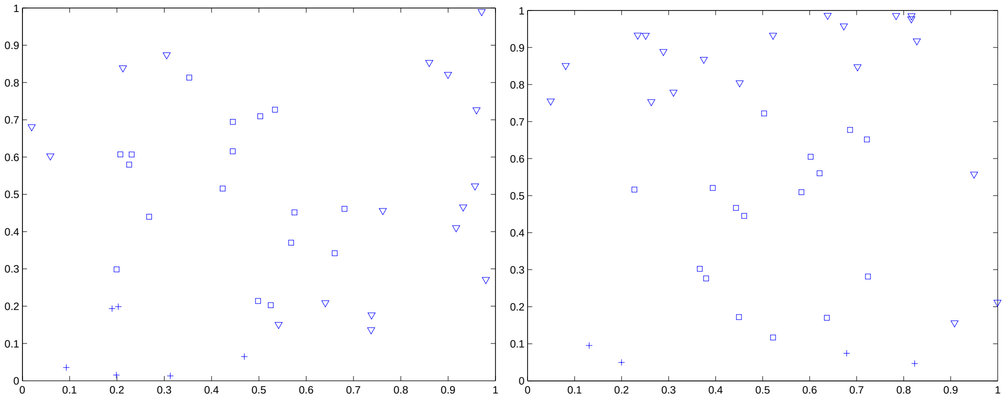
An example of a synthetic three-way classification problem is constructed to demonstrate the use of hierarchical mixture models. A similar example was used in (Neal 1997) to illustrate a model for a single group of data. The data are generated as follows: x_{ij} is generated from the uniform distribution over the interval $(0, 1)$ independently for $j = 1, \cdots, 4$. The class of the item t_i is then selected as follows:

$$t_i = \begin{cases} 0 & Dis(\boldsymbol{x}_i, \boldsymbol{x}_0) \leq 0.35; \\ 1 & \text{otherwise, if } 0.8x_{i1} + 1.8x_{i2} \leq c_0; \\ 2 & \text{otherwise.} \end{cases}$$

where $Dis(\cdot, \cdot)$ is the two-dimensional Euclidean distance of (x_{i1}, x_{i2}) from the point $\boldsymbol{x}_0 = (x_{01}, x_{02})$. Note that x_{i3} and x_{i4} have no effect on the class.

We construct a model with two mixture components with slightly different classification mechanisms, one corresponding to $\boldsymbol{x}_0 = (0.4, 0.5)$ and $c_0 = 0.6$, and the other corresponding to $\boldsymbol{x}_0 = (0.5, 0.4)$ and $c_0 = 1.0$. After the mixture component is selected, 100 cases are generated, of which 40 are used as training data and 60 as test data. Repeating the above steps ten times, we generate a data-set with 10 groups (test data 1 in Table 1). Moreover, 5 new groups of data, each with 60 cases, are generated as test data (test data 2 in Table 1). Two groups of training cases are plotted in Figure 3, corresponding to the two different mixture component classifications. The variety among the different groups is significant. This is typical of many real life classification examples, where groups would have inherent variability (e.g. different patients in a sample). As in real life, we treat the group information to the mixture model as missing.

The major computational burden is that of calculating the inverse covariance matrix in the MCMC algorithm discussed in Section 3. To make the algorithm efficient, we update the values of indicators $\boldsymbol{z}$ using 20 Gibbs sampling scans, and update the latent values $\boldsymbol{y}$ using 100 Gibbs sampling scans, since this adds little to the computation time. In each iteration, five of these combined Gibbs sampling and updates of parameters $\boldsymbol{\theta}$ were done. The algorithm is very efficient – it takes about 27 minutes on an i686 linux PC to run 1000 iterations (5000 updates of



(a) Group 1

(b) Group 2

Figure 3: Two groups of training data each with 40 training cases. Each case is plotted according to its values for x_{i1} and x_{i2}. The three different symbols stand for three different classes. Note that the figure reveals the substantial variation between the two groups.

parameters $\boldsymbol{\theta}$).

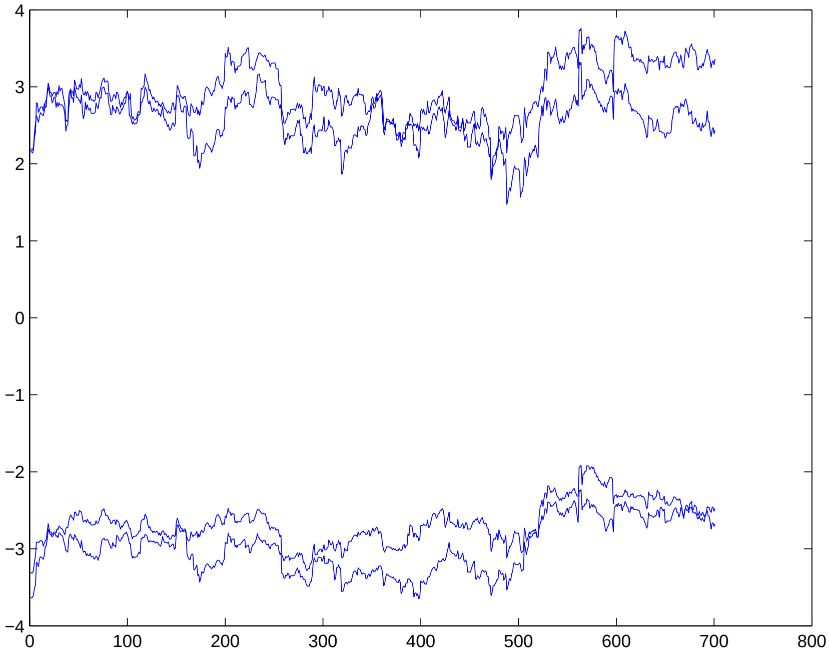


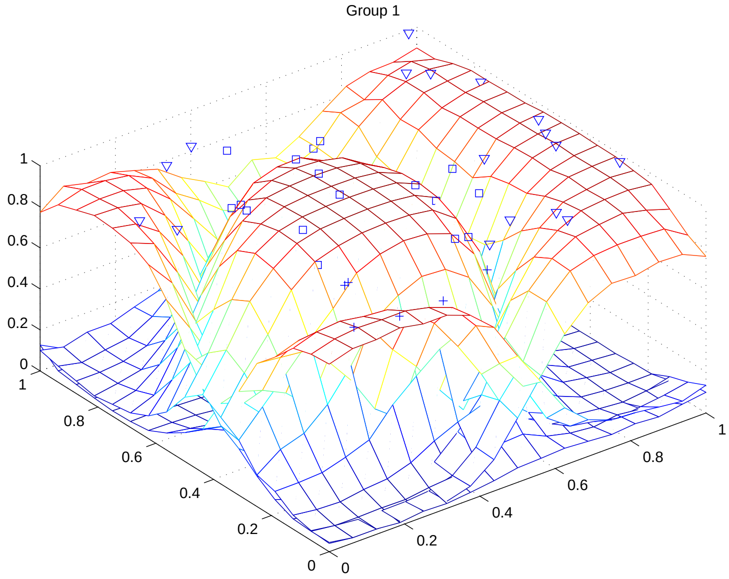
Figure 4: Progress of the values of $w_q, q = 1, \cdots, 4$ in (17) for one mixture component in MCMC simulation. The values are plotted on a log scale. The upper two curves correspond to x_{i1} and x_{i2}, and the others correspond to x_{i3} and x_{i4}.

One of the methods used to assess the convergence of the MCMC simulation is to check the values of the parameters over the course of the simulation. Figure 4 shows the values of w_1 to w_4 in (17) corresponding to one of the two mixture components. It shows that equilibrium has been achieved very rapidly, after about 40 iterations (corresponding to a total of 200 updates of parameters $\boldsymbol{\theta}$). The values of w_3 and w_4 are very small, which reflect the fact that x_{i3} and x_{i4} have no effect on the class – an example of the automatic relevance detection inherent to the Gaussian process approach to modelling. The progress of the parameters corresponding to the other mixture component is very similar.

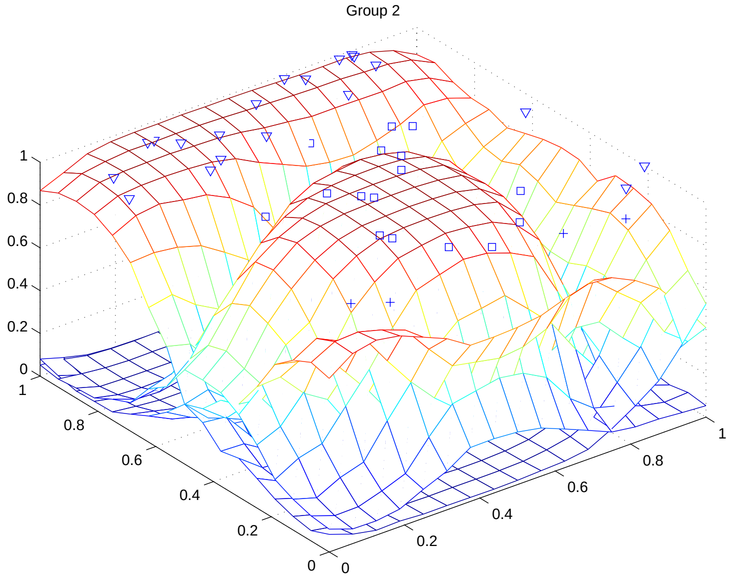
The predicted class for a test case is that corresponding to the largest predictive probability, which is calculated by (22). Both (21) and (22), involve the predictive distribution of the latent value, and this takes quite a complicated form even with the approximation (9). In practice, we can calculate the predictive mean and variance, and use the Gaussian distribution with the same mean and variance as the predictive distribution. We use 100 sample points after the process has reached equilibrium for calculating the predictive mean and variance. After this, 100 samples are generated from the Gaussian distribution, and are used to calculate the predictive probability (22). The final results are listed in Table 1, and the density functions for each class are shown in Figure 5. For test data 1, the overall classification error rate is 11.8% which is comparable to the error rate of 13% in (Neal 1997) for a single model with a single group of data set with the same sample size, 400 training data cases. Even for the previously unseen test data 2, which includes 5 new groups of test data, the error rate is about 20.1%.

Classification error rate

test data 1		test data 2	
group index	prob.	group index	prob.
1	0.1333	11	0.2333
2	0.1000	12	0.1428
3	0.1667	13	0.2784
4	0.1333	14	0.2002
5	0.1500	15	0.1516
6	0.1333		
7	0.1000		
8	0.1000		
9	0.0500		
10	0.1167		
overall	0.1183		0.2012



(a) Hierarchical model average probability density for classes – group 1



(b) Hierarchical model average probability density for classes – group 2

Figure 5: Overlaid plots of probability density $P(t = i|x)$ for each class i, plotted separately for each group. The example training data plotted previously in Figure 3 are superimposed for comparison.

4 Discussion

This paper has presented a hierarchical model which helps decompose the problems of regression and classification for batch data, and includes simulations which illustrate the approach. The hierarchical model is much more efficient for problems involving natural groups of training data. The hierarchical model deals appropriately with heterogeneity among different groups of data, and thereby improves modelling accuracy. A future extension would be to add a level of modelling to predict explictly the groups of new test data, also known as an allocation model (Fernández and Green 2000).

The use of a Bayesian approach, implemented using MCMC methods, typically leads to more robust models than optimisation-based approaches. This can be seen in the variation in the probability densities for the various groups of data.

The classification work we have presented is useful in its own right, and is an example of the use of the divide-and-conquer approach that motivates multiple model methods. It has, however, potential for general use in existing multiple model systems. Multiple models must have a blending or gating function which selects a blend of sub-models, conditional on the current state. This is essentially the same as a multi-class classification problem, so the Gaussian process classifier presented in this paper could provide a new mechanism for representing the blending function for multiple linear models, or multiple Gaussian processes. The robustness of the Bayesian MCMC approach, coupled with the hierarchical model makes this especially interesting for obtaining more realistic estimates of model variance in sparsely populated areas of the state space, as found in transient, off-equilibrium regions in dynamic systems.

References

Fernández, C. and P.J. Green (2000). Modelling spatially correlated data via mixtures: a Bayesian approach. Technical report. University of Bristol.

Gibbs, Mark N. (1997). Bayesian Gaussian Processes for Regression and Classification. PhD thesis. University of Cambridge.

Gilks, W. R. and P. Wild (1992). Adaptive rejection sampling for Gibbs sampling. *Applied Statistics* **41**, 337–348.

Holmes, C. C., D. G. T. Denison and B. K. Mallick (1999). Bayesian partitioning for classification and regression. Technical report. Dept. of Mathematics, Imperial College, London.

Johansen, T. A. and R. Babuska (2002). On multi-objective identification of Takagi-Sugeno fuzzy model parameters. In: *IFAC World Congress, Barcelona*.

Johansen, T. A. and R. Murray-Smith (1997). The operating regime approach to nonlinear modelling and control. In: *Multiple Model Approaches to Modelling and Control* (R. Murray-Smith and T. A. Johansen, Eds.). Chap. 1, pp. 3–72. Taylor and Francis, London.

Johansen, T. A., R. Shorten and R. Murray-Smith (2000). On the interpretation and identification of dynamic Takagi-Sugeno fuzzy models. *IEEE Transactions on Fuzzy Systems* **8**(3), 297–213.

Kamnik, R. and T. Bajd (1999). Force feedback in FES assisted standing-up after paraplegia. In: *Proceedings of the annual project meeting, SENSATIONS/NEUROS*. pp. 7–10.

Leith, D. and W. Leithead (1999). Analytic framework for blended multiple model systems using linear local models. *International Journal of Control* **72**(7/8), 605–619.

Lemm, J. C. (1999). Mixtures of Gaussian process priors. In: *Proc. of the Ninth International Conference on Artificial Neural Networks (ICANN99)*. IEE Conf. Pub. No. 470.

MacKay, D. J. C. (1999). Introduction to Gaussian Processes. To appear in proceedings of a NATO school.

Murray-Smith, R., T. A. Johansen and R. Shorten (1999). On transient dynamics, off-equilibrium behaviour and identification in blended multiple model structures. In: *European Control Conference, Karlsruhe, 1999*. pp. BA–14.

Neal, R. M. (1997). Monte Carlo implementation of Gaussian process models for Bayesian regression and classification. Technical Report 9702. Department of Statistics, University of Toronto.

O'Hagan, A. (1978). On curve fitting and optimal design for regression (with discussion). *Journal of the Royal Statistical Society B* **40**, 1–42.

Rasmussen, C. E. (1996). Evaluation of Gaussian Processes and other Methods for Non-Linear Regression. PhD thesis. Graduate department of Computer Science, University of Toronto.

Rasmussen, C. E. and Z. Ghahramani (2002). Infinite mixtures of Gaussian process experts. In: *Advances in Neural Information Processing Systems 14* (Z. Ghahramani T. Dietterich, S. Becker, Ed.). MIT Press.

Shi, J. Q., R. Murray-Smith and D. M. Titterington (2002). Hierarchical Gaussian process mixtures for regression. Technical Report TR-2002-107. University of Glasgow, Scotland, UK.

Shorten, R., R. Murray-Smith, R. Bjørgan and H. Gollee (1999). On the interpretation of local models in blended multiple model structures. *International Journal of Control* **72**(7/8), 620–628.

Titterington, D.M., A.F.M. Smith and U.E. Makov (1985). *Statistical Analysis of Finite Mixture Distributions*. John Wiley & Sons. Chichester.

Williams, C. K. I. (1998). Prediction with Gaussian processes: From linear regression to linear prediction and beyond. In: *Learning and Inference in Graphical Models* (M. I. Jordan, Ed.). pp. 599–621. Kluwer.

Williams, C. K. I. and C. E. Rasmussen (1996). Gaussian processes for regression. In: *Neural Information Processing Systems - 8*. MIT press. Cambridge, MA. pp. 514–520.

Williams, C. K. I. and D. Barber (1998). Bayesian classification with Gaussian Processes. *IEEE Trans Pattern Analysis and Machine Intelligence* **20**(12), 1342–1351.

5 Acknowledgements

The authors gratefully acknowledge support from EPSRC grant *Modern statistical approaches to off-equilibrium modelling for nonlinear system control* GR/M76379/01. RM-S also acknowledges the support of the *Multi-Agent Control* Research Training Network supported by EC TMR grant HPRN-CT-1999-00107. We would like to thank R. Kamnik, and T. Bajd of the University of Ljubljana for allowing us to use the FES data.