8-15-2014

Mixtures of Self-Modelling Regressions

Rhonda D. Szczesniak
Cincinnati Children's Hospital Medical Center

Kert Viele
University of Kentucky, kert.viele@uky.edu

Robin L. Cooper
University of Kentucky, rlcoop1@uky.edu

Click here to let us know how access to this document benefits you.
Mixtures of Self-Modelling Regressions

Notes/Citation Information


This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Digital Object Identifier (DOI)
https://doi.org/10.4172/2155-6180.S12-003
Mixtures of Self-Modelling Regressions

Rhonda D Szczesniak1*, Kert Viele23 and Robin L Cooper4

1Division of Biostatistics and Epidemiology, Cincinnati Children’s Hospital Medical Center, USA
2Berry Consultants, LLC, USA
3Department of Statistics, University of Kentucky, USA
4Department of Biology, University of Kentucky, USA

Abstract

A shape invariant model for functions $f_i$, ..., $f_K$ specifies that each individual function $f_i$ can be related to a common shape function $g$ through the relation $f_i(x) = a_i g(c_i x + d_i) + b_i$. We consider a flexible mixture model that allows multiple shape functions $g_1, ..., g_K$, where each $f_i$ is a shape invariant transformation of one of those $g_k$. We derive an MCMC algorithm for fitting the model using Bayesian Adaptive Regression Splines (BARS), propose a strategy to improve its mixing properties and utilize existing model selection criteria in semiparametric mixtures to select the number of distinct shape functions. We discuss some of the computational difficulties that arise. The method is illustrated using synaptic transmission data, where the groups of functions may indicate different active zones in a synapse.

Introduction

Self-Modelling regressions form a class of models [1,2] for functional data observed for many individuals. We observe data, where the subject specific functions $f_i(x)$ (sometimes referred to as traces) have the form where $g$ is a base function and $\theta_i$ is a subject specific parameter which specifies a transformation connecting $g$ and the functions $f_i$. A variety of choices is available for this transformation class which unites the subject specific functions $f_i$ to $g$. One of the first examples in the literature, which we will address here, is the shape invariant model, referred to as SIM (1), where $\theta_i = (a_i, b_i, c_i, d_i)$ are called self-modeling coefficients and

$$f_i(x) = a_i g(c_i x + d_i) + b_i$$

The self-modeling coefficients result in $f_i$ being an affine transformation of $g$ in the $x$ and $y$ axes. Other types of transformations have been shown which use a Bayesian warping function method [3]. Altman and Villarreal [4] have developed the SIM where interest centers on variation in the self-modeling coefficients, using nonlinear mixed effects [5] and p-splines to model the $\theta_i$ parameters. More recent work involved inferential comparisons on $\theta_i$ in the SIM setting [6]. In this work, our primary objectives are the following:

1. To consider mixtures of shape invariant models (MSIMs) where there is more than one underlying shape function $g$. Unknown to the investigator, suppose the subject specific functions may be grouped so that one group is defined by shape invariant transformations of an underlying $g_k$, another group is defined by shape invariant transformation of an underlying $g_k$, and so on. A motivating example follows below.

2. To utilize Bayesian Adaptive Regression Splines (BARS) [7-9] to estimate the underlying shape functions $g_k$ for $k=1, ..., K$. BARS has been shown to provide parsimonious fits (e.g. fewer knots) of complicated functions. Several computational challenges arise in these endeavors which are described in Implementation.

In our motivating example, data are obtained from a recording of synaptic transmission at a neuromuscular junction (NMJ) from a crayfish (Figure 1a). The motor nerve is repeatedly stimulated and the synaptic response is monitored as described previously in work by two of the authors [10]. With low frequency stimulation of this low output NMJ, occasionally an evoked response is observed. In the sample shown, 61 events occurred in 1000 stimulations. After some data cleaning [11] our data of interest is the 61 functions describing voltage over a dense series of time points.

These functions vary in terms of their heights, widths, and peak locations, but in fact do all have similar shapes as shown by the aligned curves which result from fitting a SIM model to all the functions (Figure 1b). The details of this fit, which simply assumed known knots for a
Mixtures of Shape Invariant Models

Prior and likelihood structure

Suppose we observe \( I \) individuals and for each individual we observe data pairs \((x_{ij}, y_{ij})\) for \( i = 1, \ldots, I \) and \( j = 1, \ldots, J \). We also assume there exist underlying shape functions \( g_1, \ldots, g_N \) such that the \( i \)th individual follows shape \( g_i \) with probability \( p_i \), where \( p = (p_1, \ldots, p_N) \) is a vector of probabilities. Let \( z_i \) be a group indicator where \( z_i = k \) indicates that the \( i \)th individual follows the \( k \)th function and that \( \theta = (a,b,c,d) \) are the self-modeling coefficients for the \( i \)th individual. Finally \( \sigma^2 \) is the error variance. Where possible, we place conditionally conjugate priors on the parameters to ensure an easier MCMC implementation. In what follows \( g \) refers to the collection of \( g_1, \ldots, g_N \) and \( \theta \) refers to the collection of \( \theta_1, \ldots, \theta_N \) and \( N(\mu, \sigma^2) \) refers to a normal density with mean \( \mu \) and variance \( \sigma^2 \) evaluated at \( x \), with similar notation for other densities. The general prior and likelihood structure is below. We will describe the priors on the \( \theta \) and \( g_i \) separately.

\[
\pi(\theta, g, \sigma^2, p) = \left[ \prod_{i=1}^I \pi(\theta_i) \right] \left[ \prod_{k=1}^K \pi(g_k) \right] \pi(\sigma^2) \pi(p)
\]

\[
\pi(\sigma^2) = \text{InvGamma}(\alpha_2, \beta_2 \sigma^2)
\]

\[
\pi(p) = \text{Dir}(\lambda_1, \ldots, \lambda_K | p_1, \ldots, p_K)
\]

\[
z_i \mid \theta_i, g_i , \sigma^2, p \sim \text{Multinomial}(1, p_1, \ldots, p_K)
\]

For a more thorough description of curve registration techniques and other, potentially helpful transformations for improving MCMC mixing, we refer the reader to the original paper by Ramsay and Li [14] and Gilks and Roberts [15], respectively.

To see the benefit of this transformation by example, let \( g \) be a triangular "tent function" where \( g(x) = 0 \) outside \((0.2,0.6)\), \( g(x) \) increases linearly from 0 to 1 over \((0.2,0.4)\) and decreases linearly from 1 to 0 over \((0.4,0.6)\). We generated data using \( f(x) = g(0.5x - 0.25) \) for \( x = (0, 0.01, 0.02, \ldots, 2) \) and \( \text{N}(0,1) \) error. The peak of \( g \) occurs at 0.4 and the corresponding peak of \( f \) occurs at \( m = 1.3 \). Keeping everything fixed at

\[
f(x) = a g_k(x) + b
\]

\[
Y_j | x_j, z_j, \theta_j, g_j, \sigma^2, p \sim \text{N}(f(x_j), \sigma^2)
\]
the true value except \( c \) and \( d \), Figure 2a shows the loglikelihood over \((c,d)\). As can be seen, there is a distinct correlation between \( c \) and \( d \) which limits the mixing of the MCMC chain. In contrast, Figure 2b shows the loglikelihood as a function of \( c \) and \( m \), which provides a much better structure for MCMC.

Of course, this transformation is limited to \( g \) with a single peak, which is not always true. Thus, this transformation will not produce an improvement in general, or other transformations may be useful in other contexts. For our synaptic transmission data, we have found this transformation to be quite useful. Thus, for data with a dominant peak:

\[
\pi = \begin{pmatrix} a, h, c, m \end{pmatrix} \sim N(μ_0, \Sigma_0, k(a, h)) \left[ \text{Gamma}(α, β)(c) \right] \left[ U(0,1)(m) \right]
\]

**Identifiability**

Identifiability is a pervasive issue in self-modeling regressions\,[2,4]. Suppose for example we have a particular underlying shape function \( g(x) \) and a set of coefficients \((a,b,c,d)\). One could achieve the same \( f \) by taking \( g'(x) = 2g(x) \) and taking \( a' = a/2 \). Generally, one can take any self-modeling transformation of \( g \) and then adjust \((a,b,c,d)\) appropriately to produce the same \( f \). In the original Lawton paper\,[1], this problem was addressed by forcing \( g \) to have particular properties (such as a maximum of 1 and minimum of 0). However, in the Bayesian formulation such constraints would require any estimation method to be significantly altered. BARS, which we utilize here, is not obviously adjustable to handle constraints on the function.

An alternative method of forcing identifiability is to pick an individual function \( f \) and force its corresponding \((a,b,c,d)\) to be \((1,0,1,0)\). This method is utilized in previous work\,[10]. This forces the scaling on \( g \) to follow the chosen function but allows BARS to run un constrained.

The central theme here is that in the self-modeling paradigm you are really estimating relative shifts and scalings among the functions.

Unfortunately, in the mixture setup this is not feasible, as there are multiple shape functions. Since we do not know a priori which functions, \( f_1, \ldots, f_L \), belong to which group (e.g. which are self-modeling versions of \( g_1 \), which are self-modeling versions of \( g_3 \), etc.) we cannot select functions to use as “anchors” in each group. Thus the simplest option seems to be to leave a nonidentified model. Altman and Villareal\,[4] also use an unconstrained formulation without difficulty, though from a frequentist standpoint.

Note, however, that our central inferential goals are the assignment of functions to groups, the estimation of the individual functions, and the underlying shape of the \( g_{1}, \ldots, g_{K} \) functions. The first two are identifiable, and \( g_{1}, \ldots, g_{K} \) are identifiable up to self modeling versions. As with any mixture model, there is a secondary identifiability problem in that the component labels may be switched, but component label switching\,[14] has not been an issue in the analyses we performed. Although not addressed in this work, there is a third identifiability problem in mixture models arising from having an unnecessarily large number of mixtures when a smaller number of mixtures is actually sufficient.

**Implementation**

We implemented the model in Section 2 using an MCMC scheme. Some parameters are of course easier to update than others.

**Sampling scheme**

- **Updating \( σ^2 \) and \( p \):** Both \( σ^2 \) and \( p \) have straight forward conjugate priors conditional on the rest of the parameters.

\[
\pi(\sigma^2, p | \text{rest}) = \text{Dir}(λ_1 + \# Z_1 = 1, \ldots, λ_K + \# Z_K = K) (p_1, \ldots, p_K)
\]

\[
\pi(σ^2 | \text{rest}) = \text{InvGamma}(α_x + \frac{N}{2}, β_x + \sum_{i=1}^N e_i^2/2),
\]

where

\[
e_i^2 = (y_i - \hat{y}_i)^2 \quad N = \sum_{i=1}^N q_i
\]

- **Shape functions \( g_1, \ldots, g_K \):** Conditional on the remaining variables, each \( g_k \) is conditionally independent, with each \( g_k \) being updated separately using only those data functions with \( z = k \).

Each \( g_k \) is defined by its corresponding knot set \( ξ_k \) and spline coefficients \( β_k \). In the original BARS algorithm\,[7], updating was done entirely on \( ξ \) using BIC with the maximum likelihood estimate of \( β \). While we update \( ξ \) using the BARS updating, we actually draw \( ξ \) via \( β | \text{rest} \) and \( β | \xi \), rest. Our reason for drawing \( β \) values is that, unlike the original work\,[7], we have additional parameters (specifically \( z \) and \( ξ \)) that are easier to simulate if \( β \) is fixed.

- **Updating \( Z \):** Updating each \( Z \) also requires moving functions between groups, as there are subject specific parameters \( θ \) for each variable. In the MCMC scheme one must take into account that \((a,b,c,d)\) have different interpretations depending on which group the function belongs to (e.g. the outcome of \( Z \)). If \( g_1 \) and \( g_2 \) have different peaks, for example, then a perfect fit for \( g_1 \) would require a much different value of \( α \) than a perfect fit for \( g_2 \).

Ideally, we would like to avoid this problem and integrate the self-modeling coefficients out to find

\[
Pr(Z_i = k | g_1, \ldots, g_K, p, σ^2) = \int Pr(Z_i = k | g_1, \ldots, g_K, p, σ^2, θ)π(θ)dθ
\]

Unfortunately, it is not computationally efficient to find this
integral nor is it efficient to approximate it (for example finding the MLE of $\theta$ would require numerical maximization). Thus in our sampler we utilize $\theta_i$ (different $\theta$ for each $k$). To draw a new $Z_i$, note

$$
Pr(Z_i = k | g_1, \ldots, g_m, p, \sigma^2, \theta) = \frac{p_k}{\sum_{k=1}^{M} \prod_{i=1}^{N} f(x_i) \sigma^2(y_i)}
$$

where $f(x) = a_k g_k(x_i) + b_k$.

**Updating the self-modeling coefficients:** Finally, the self-modeling coefficients $(a,b)$ (recall in the sampler we have separate self-modeling coefficients $\theta_i$ for each $i$ and $k$) are updated straightforwardly, as they are regression parameters with conjugate priors. The $(c,m)$ coefficients are updated using a Metropolis-Hastings random walk scheme.

**Singularities in the BARS updating**

In the original BARS implementation, proposed knot sets which resulted in singular design matrices (loosely, too many knots in an area with too few data points), were discarded [16,17]; these proposals may be viewed simply as a data dependent prior. In the self-modeling setup, there is an additional complication. In one iteration we may achieve a nonsingular design matrix, but as the self-modeling coefficients $\theta$ are adjusted, this moves $m$ defined in Section 2.1.2 and may result in a singularity in the design matrix. We additionally reject these moves out of hand in the sampler. At present, these rejections in the sampler are not a frequent occurrence, but seems an unavoidable problem as long as BIC is used in the BARS steps.

**Starting values**

To initialize the chain, all $f$ functions were first aligned to the largest function (the trace with the largest difference between the maximum and minimum values) using a Procrustes registration [14]. We then computed the mean square error (MSE) between each aligned trace and the largest trace. Focusing on a two-component mixture, the maximum and minimum values) using a Procrustes registration [14].

The amount of burn in necessary for Gibbs sampling in the MSIM setting depends on data-specific and method-specific factors. Wallstrom et al. [17] recommend up to 20,000 burn-in iterations for the BARS implementation with normally-distributed data. We found that additional iterations were necessary to achieve proper burn in for the self-modeling coefficients.

**Improving acceptance rates**

It is important that the sampling mechanism provide a sufficient amount of switching in order to explore the state space well. Typically, one would choose a reversible-jump mechanism to facilitate this switching, but in most RJMCMC scenarios it is easier to sample from the posterior distribution within each subgroup (i.e. mixture component). The aforementioned sampling mechanism in Section 3 uses metropolis random walks within each subgroup, which may lead to lower acceptance rates, thereby lowering the degree of switching between groups.

To facilitate switching between groups, we propose using the approximating posterior distributions of the self-modeling coefficients in $\theta$ from our previous MCMC implementation in Section 3 to conduct a second MCMC. In this second MCMC, acceptance rates may be improved as follows. Let $A_{i,k\rightarrow}$ be the probability of the $i$th trace moving from the current state $M_i$ to the candidate state $M_k$. For ease of description, we will now omit notation for trace $i$. The acceptance probability for moving from the current state ($k$) to the candidate state ($k'$) is

$$
a_{k\rightarrow k'} = \frac{\pi(M_k\mid a_{i,k}, b_{i,k}, c_{i,k}, d_{i,k}) \pi(M_k) \pi(y\mid a_{i,k}, b_{i,k}, c_{i,k}, d_{i,k})}{\pi(M_{k'}\mid a_{i,k'}, b_{i,k'}, c_{i,k'}, d_{i,k'}) \pi(M_{k'}) \pi(y\mid a_{i,k'}, b_{i,k'}, c_{i,k'}, d_{i,k'})} \frac{p_k}{p_{k'}}
$$

where in general notation for state $m,n (M_{m,n})$ is the prior probability that the trace has underlying shape function $g_n$. $f_i(|\cdot)$ refers to coefficients $(a,b,c,d)$ being evaluated at the multivariate normal distribution with respective mean vector $p_{i,n}$ and covariance matrix $\Sigma_n$ obtained from the complete conditional distributions in the first MCMC implementation; $p_n$ is the empirical probability that the trace belonged with shape function $g_n$ in the first MCMC; $\pi(y|a,b,c,d,M_{m,n})$ is the likelihood for the trace under shape function $g_m$. The $\pi(a_{i,k}, b_{i,k}, c_{i,k}, d_{i,k}|M_{m,n})$ corresponds to coefficients $(a,b,c,d)$.  

**Assessing Implementation Results**

**Posterior inference**

At the end of the MCMC run we have a set of $M$ iterations. For each $n=1,\ldots,M$, the $m$th iteration contains functions $g_{m1}^{(m)}, \ldots, g_{mN}^{(m)}$, (expressed in a spline basis), mixing proportions $\pi^{(m)}$, an error standard deviation and component assignments $z_{i}^{(m)}$, $z_{j}^{(m)}$ for each individual function. Finally, for each $i$ we have self-modeling coefficients $\theta_i^{(m)}$. We assess the results in terms of the identical pieces of the sampler.

1. We directly have the mixing proportions $\pi$ between the different functions $g_i$. We also compute the estimated posterior probability each individual trace belongs to each group by finding the observed proportions of times $z_i^{(m)} = k$ for each $k$. Thus, we acquire overall proportions of functions in each group and the estimated posterior probabilities each individual function belongs to each individual $g_i$.

2. To assess the fit of the individual functions, we computed the estimated $f(t)$ by constructing for each iteration and each trace

$$
\hat{f}_i(t) = \frac{1}{M} \sum_{m=1}^{M} \hat{f}_{i,m}^{(m)}(t)
$$

In the plots that are constructed from the simulation studies and EPSPs application in Section 5, we plot a selection of the $\hat{f}_i(t)$ functions (iterations $m = 100,200,\ldots$) and the overall average $\hat{f}(t)$.

To estimate each $g_i$ we must remember that each $g_i$ is allowed to drift in the sampler, and thus the resulting $g_i^{(m)}$ iterations must be aligned in shape before producing an estimate. Thus, we begin by taking the functions $g_i^{(m)}$ across the iterations and aligning them to each other. This was done via a Procrustes registration as in previous work [10]. The base shape function of this alignment was used as the estimate for $g_i$.
being evaluated using priors under $M_k$, which can be simplified as $\pi(\alpha, \beta, c, d | M_k) = \pi(\alpha | M_k) \pi(c | M_k) \pi(d | M_k)$; prior $\pi(d | M_k)$ is an induced prior that we compute via transformations. The algorithm for computing the acceptance probabilities is outlined in Supplementary Information.

**Determining the number of mixtures**

One outstanding issue still to be addressed is inferentially determining the number of groups present in the data. When making this determination, BIC is intuitively a reasonable choice; however, regularity conditions do not hold in the mixture model setting [18]. Furthermore, BIC is not reliable in the mixture model settings with smaller sample sizes. Pilla and Charnigo [19] proposed a flexible information criterion (FLIC) specific to selecting the number of components in semiparametric mixture models, which performs better than BIC in settings where the mixture components are not easily distinguishable and better than AIC when sample sizes are large. We adapted the FLIC implementation used previously for fitting a normal mixture model to a distribution of birth weight [20] as follows. For a mixture of self-modeling regressions with components $k=1, \ldots, K$

$$FLIC_k = -2 \log L_k + 2 \log(N)^{0.5} \nu,$$

where, $L_k$ is the log likelihood of the data evaluated at the posterior mean estimates for each function $f_i; i=1, \ldots, I$. $N$ is the total number of observations defined in Equation (1). The number of free parameters in the model ($\nu$) is the number of shape functions $g_k$ (note that we do not consider the number of parameters in the knot and spline coefficient vectors since we can use the posterior mean $\hat{g}_i$ from the sampler, post drift alignment, as our estimate for $g_k$), plus the number of undetermined mixing proportions ($K-1$ free parameters), the total number of self-modeling coefficients in the $4 \times 1$ vector $\theta$ for all $I$ subjects, and the common variance parameter $\sigma^2$. The bivariate ratio function $B(\nu, \delta)$ and penalty statistic $\delta$ have been defined previously [19]. For each MSIM considered, we can compute the fraction of between-component variability to within-component variability across the $x$ values (or time points) and use these results to compute the average fraction of between-component variability to within-component variability (see formula in Supplementary Information).

**Results**

The MCMC implementation and post processing described in Sections 3–4 were executed for simulation studies and the motivating EPSPs example using R [21].

**Simulations**

We simulated data to identify how well the algorithm discriminated two separate, but similar functions. In each sample dataset we used the following underlying shape functions:

$$g_1(x) = 0.2 \exp[-20(x-0.2)^2]$$

$$g_2(x) = (\exp[-0.7x] - \exp[-12x])I_{(x)}$$

These functions are similar to some parametric models used in the biology literature for EPSPs, and are quite similar in shape. Figure 3 shows the two underlying functions. The two functions are similar in shape, but they are clearly not identical. In other words, these two shape functions are clearly not self-modeling versions of each other. Although $g_1$ is defined as a symmetric function, its alignment with $g_2$ makes it appear asymmetric over the range of $x$. The function $g_2$ is not differentiable at $x=0$ (note self-modeling versions with a slight shift are quite similar, so this cusp could appear at a different $x$ value in the estimate, see the discussion of identifiability in Section 2.2), which creates a point where multiple knots are desirable. This is a situation where adaptive knot splines are more effective than using known knots.

Each sample dataset consisted of 100 functions, 50 generated from...
self-modeling versions of $g_1$ and 50 from self-modeling versions of $g_2$. The self-modeling coefficients $(a,b,c,d)$ were randomly chosen for each function in each simulation to provide a range of overlapping functions. The range of values from which $a,b,c,d$ were selected consisted of $(3.0,7.0), (-0.15,0.10), (0.15,0.90),$ and $(-0.06,0.80)$, respectively. The $x$ values for each function correspond to 121 equally-spaced points in the interval $(0,1)$. Forty simulated datasets of 100 functions were generated, 20 of which had an error standard deviation of $\sigma=0.15$ while the remaining 20 had an error standard deviation of $\sigma=0.05$. One sample dataset (for $\sigma=0.05$) is shown in Figure 4a. Figure 4b shows the same dataset color coded by the appropriate $\gamma$ function. It is certainly not obvious from the data alone which $\gamma$ is appropriate. The first- and second-stage MCMC algorithms from Equation (3) were run on each simulated dataset. The first-stage sampler was run for 80,000 iterations with the first 10,000 iterations removed. For each simulated dataset we computed the acceptance rates. In lower noise ($\sigma=0.05$) data, switching improved in the second run. Of the 20 datasets generated with lower noise $\sigma=0.05$, all but 3 had all 100 functions classified correctly. For those 3 datasets, 2 had 99 functions correctly classified and the other had 97 functions correctly classified. For the 20 datasets with $\sigma=0.15$, there was more variability because of the increased noise but there were still high correct classification rates, ranging from a low of 77 classified correctly to a high of 94.

An improvement in switching was defined as moving from the current state to the candidate state more often during post-burnin iterations in the second MCMC, compared to movement in the first MCMC. For higher noise ($\sigma=0.15$) data, all functions had improved acceptance rates. In lower noise ($\sigma=0.05$) data, switching improved in all but 3 datasets. For those 3 datasets, there were 4, 7, and 25 functions that did not improve their acceptance rates in the second run. We calculated FLIC from Equation [4] for each simulation using one-group and two-group results. All lower-noise simulations favored including two distinct shape functions $g_1$ and $g_2$ over a single shape $g$. Eighteen of the higher-noise datasets favored the two-group model while 2 of the datasets had FLIC results favoring the single group model.

Classifying firings from synaptic transmission data

Of the 61 firings, 31 were classified to group 1 over 50% of the time while the other 30 were classified to group 2 over 50% of the time (to make sure the algorithm simply didn’t split the 61 functions in half...
arbitrarily, we also fit subsets of the functions and found consistent assignments). Figure 6a is identical to Figure 1a except the functions are color coded into group 1 (red) and group 2 (blue). Acceptance probabilities were increased in the second MCMC run, but group membership changed for only 2 of the 61 firings from the first run to the second run.

There is no obvious pattern to the assignments in Figure 6a (e.g. group 1 traces are not generally higher, wider, etc.). When we construct the plot analogous to the aligned data in Figure 1b, we do see a pattern emerge. In Figure 6b, the central difference in the groups appears near the peak of the functions assigned (here meaning greater than 0.5 posterior probability) to each group. Thus the diagonal panels in this figure correspond to good fits (e.g. the functions assigned to group 1 with \( g_1 \) overlaid, and the functions assigned to group 2 with \( g_2 \) overlaid); the off diagonal panels show the differences between the groups (e.g. the function assigned to group 1 with \( g_2 \) overlaid, and the functions assigned to group 2 with \( g_1 \) overlaid). We compared the model results from the FLIC for a single shape versus two shapes and found the model with two shapes was more favorable. Again, the two different shapes are apparent in the graph.

**Figure 7:** Group assignments and fitted firings. The four panels show the functions assigned to each group versus the fitted curves for \( g_1 \) and \( g_2 \) from the synaptic transmission data. The diagonal panels (a) and (d) show the functions assigned the corresponding \( g \) (e.g. the functions assigned to group 1 with \( g_1 \) overlaid, and the functions assigned to group 2 with \( g_2 \) overlaid). In contrast, the off-diagonal panels (b) and (c) correspond to the functions overlaid with the “other” \( g \) (so the functions assigned to group 1 with \( g_2 \) overlaid, and the functions assigned to group 2 with \( g_1 \) overlaid). This illustrates the differences in structure between \( g_1 \) and \( g_2 \).

**Discussion**

Fitting an MSIM allows for multiple shapes to be present in a sample of functions. In substantive contexts like the synaptic transmission data, finding these groups has a direct impact on questions like the number of active zones in a synapse. The algorithm proposed in this paper behaves well in simulations and provides results for the synaptic transmission data consistent with biological expectations.

Recent work by Zhang and Telesca [22] focuses on clustering and general registration of functional data where the number of knots and their locations are specified a priori and common to all underlying shapes. They show this approach works well for clustering growth data and gene expression curves. Our application to EPSPs and simulation studies address settings where the underlying shape functions are quite similar but perhaps one or more of the shape functions are not differentiable over the entire interval of interest. In these settings, we have found it advantageous to use BARS, compared to using known knots, as adaptive methods can include multiple knots to estimate cusps or other sharp changes in curvature.

The degree of switching (i.e. changes in acceptance rates) generally increased from the first MCMC to the second MCMC but appeared to depend on overall noise in the data. These results are consistent with existing literature on tuning proposal distributions [23]. It is possible that other methods of ”data-driven” tuning which incorporate overall variance in the MSIM may be necessary to further improve acceptance rates in the RJMCMC scheme [24].

Substantively, the finding of groups is expected, as such a finding can be directly linked to the underlying structural entities which are responsible for efficacy in chemical synaptic transmission in general [12,25,26]. However, if one fits a model with two groups, then two groups will appear in the results. While we were able to adapt an existing information criterion for this assessment [19] and obtain reasonable results, in the future it will be important to formally develop an inferential procedure for MSIMs. Fortunately for the synaptic transmission data, simpler mixture models have also detected two groups [12].

In univariate Gaussian mixture models, there are various model selection criteria that could be readily supplied [18,27] or one could use reversible-jump samplers to move between models with differing numbers of components in a manner similar to a previously-described approach [28]. In our context it is unclear how to implement a model selection criteria that requires the number of parameters in the model, as the spline formulation includes the number of knots as one of the unknown parameters. In this work, we substitute these unknown parameters with the K parameters corresponding to estimating \( \beta_k \) for \( k=1,…,K \). Even simplistic criteria such as investigating the MSE between functions, or more precisely the use of \( \chi^2 \) or noncentral \( \chi^2 \) distributions, is complicated by the differing scaling produced by the self-modeling coefficients for each function. It would be of interest in future work to formally incorporate spline complexity in the fit statistic evaluation.

**Acknowledgement**

This research was supported by NCCR(NIH) Grant P20 RR16481. The authors are grateful to Richard Charnigo, PhD, for sharing implementation code for the flexible information criterion. The authors thank the reviewers for their comments, which have substantially improved the paper.

**References**


