On nonlinearity in neural encoding models applied to the primary visual cortex

Diego Vidaurre, Concha Bielza and Pedro Larrañaga

Computational Intelligence Group
Departamento de Inteligencia Artificial, Universidad Politécnica de Madrid
diego.vidaurre@fi.upm.es
mcbielza@fi.upm.es, pedro.larranaga@fi.upm.es

Abstract. Within the regression framework, we show how different levels of nonlinearity influence the instantaneous firing rate prediction of single neurons. Nonlinearity can be achieved in several ways. In particular, we can enrich the predictor set with basis expansions of the input variables (enlarging the number of inputs) or train a simple but different model for each area of the data domain. Spline-based models are popular within the first category. Kernel smoothing methods fall into the second category. Whereas the first choice is useful for globally characterizing complex functions, the second is very handy for temporal data and is able to include inner-state subject variations. Also, interactions among stimuli are considered. We compare state-of-the-art firing rate prediction methods with some more sophisticated spline-based nonlinear methods: multivariate adaptive regression splines and sparse additive models. We also study the impact of kernel smoothing. Our goal is to demonstrate that appropriate nonlinearity treatment can greatly improve the results. We test our hypothesis on both synthetic data and real neuronal recordings in cat primary visual cortex, giving a plausible explanation of the results from a biological perspective.

1 Introduction

In neuroscience, encoding is the task of studying the spike firing rate of a neuron or ensemble of neurons in response to some stimuli. The prediction of neural firing patterns from external, dynamic stimuli is an important task for understanding neuronal behavior (Brown et al, 2004; Kass et al, 2005). It is well known that pure linear models often fail to infer spiking responses (Machens et al, 2004). In general, this may depend on the type of stimulus, the specific physiology of the observed neurons and the species under study. However, some amount of nonlinearity (often high) is always to be expected.

We consider models where the response is the spike firing rate of a single neuron at each time point, and the input variables or predictors are the previous stimuli within a certain time window. Typically, nonlinearity can be introduced after (over) a linear temporal filter on the time-varying stimulus, like the linear-nonlinear Poisson model (Brenner et al, 2000; Paninski, 2004;

Pillow and Simoncelli, 2006), or before the linear temporal filter (Ahrens et al, 2008). In the regression framework, this linear temporal filter is equivalent to the vector of regression coefficients. By establishing a static nonlinear transformation on some linear combination of the inputs, the first approach can capture the particular nonlinearity of the actual spike generation for example. These models are of no concern to this paper. We focus on the second approach, which builds nonlinearity separately on each input before applying the linear temporal filter, thus dealing with nonlinear dendritic behavior and other preliminary nonlinear neuron processes. Both approaches consider only an additive (linear) contribution of the inputs, i.e., they do not include interactions among different inputs.

Our starting points are the models introduced by Ahrens et al (2008): the bilinear model and the fullrank model. Both models make use of a basis expansion of the predictors (Schumaker, 2007) to achieve nonlinearity, ignoring any possible interactions among different stimuli. Therefore, these models address the preliminary neuron processes of the second approach mentioned above.

We intend to reach more complex nonlinear relations, which will presumably help to deal with the complex within-neuron processes. The techniques used in this paper, whose building blocks are well known to the statistics community, generalize the linear regression setting to accommodate nonlinearity by expanding the original set of variables, much like the bilinear and fullrank models.

On the other hand, many of the methods in the literature ignore the internal variation of the subject. Once a model (parameterized by a set of parameters) is obtained, it will remain unchanged for all future inferences. However, some studies reveal that several neural systems can vary the spiking pattern; see for example (Bezdudnaya et al, 2006; Haider et al, 2007). Here, we also study how to generate ad-hoc, adaptive model parameters for each time point and what impact it has on the model performance. We make use of kernel smoothing (Loader, 1999), a technique supported by a solid theoretical background.

The proposed local models are to some extent related to models that include spike-history terms, which also consider the internal variation of the subject by making the current response to be dependent on previous responses. Local models include internal variation in a more general manner, without expressing variation just by means of previous responses. Escola et al (2011) recently propose to model multistate neurons by hidden Markov models, where each state corresponds to a different generalized linear model. Although this is a successful idea, local models are easier to obtain and, since subject variation is in this paper defined on a continuum, we do not need to beforehand fix the number of states.

The rest of the paper is organized as follows. In Section 2, we set out the notation, formalize the basic concepts and revisit the bilinear and the fullrank models. In Section 3, we briefly survey some important concepts about nonlinear models and describe the models to be used in the experimental part. In Section 4, we present the results on several synthetic data sets. In Section 5, we describe the results on a real data set of neuronal recordings in cat primary visual cortex. Finally, in Section 6, we discuss the results and outline some final conclusions.

2 Setting and preliminary methods

We consider that, at each time point $t \in \{1, ..., T\}$, we have a d-dimensional stimulus $\boldsymbol{s}(t)_{t=1}^{T}$ and a single-neuron response $r(t)_{t=1}^{T}$ to these stimuli. The objective is to predict each response $r(t)$ from previous consecutive stimuli $\boldsymbol{s}(t-i)_{i=1}^{p}$ Let $r(t)$ represent the number of spikes in a time slice centered at t, i.e., the firing rate at t. This is the well-known peristimulus time histogram (PSTH) (Gerstein and Perkel, 1969). In this paper, $r(t)$ is measured as the average spike count at time point t over B trials,

$$r(t) = B^{-1} \sum_{b=1}^{B} r_b(t), \tag{1}$$

where $r_b(t)$ is the measured spike count at time t for trial b. This way, the model omits membrane-memory effects and erratic bursting. Hence, we can model $r(t)$ as a function only of previous stimuli plus some noise:

$$r(t) = g(\boldsymbol{s}(t-i)_{i=1}^{p}) + \epsilon(t), \tag{2}$$

where $g(\cdot) : \mathbb{R}^p \to \mathbb{R}$ is some nonlinear function, p is a parameter indicating the number of past stimuli influencing the current response and $\epsilon(t)$ is the noise term. We model the noise as

$$\epsilon(t) \sim N(0, \sigma^2(t)). \tag{3}$$

Noise stands for variability in the response that is not explained by the stimuli, and may either be given by internal processes or be purely random. Provided that $r(t)_{t=1}^{T}$ is trial-averaged, the Gaussian noise assumption is reasonable (Averbeck et al, 2006; Ahrens et al, 2008).

The basis for our comparisons is two basic nonlinear models devised by Ahrens et al (2008) that lean on the linear regression framework. These are the bilinear model and the fullrank model. The *bilinear* model computes the estimated response $\hat{r}(t)$ as

$$\hat{r}(t) = \hat{\mu} + \sum_{i=1}^{p} \sum_{j=1}^{d} \beta_{ij} f(s_j(t-i)), \tag{4}$$

where $s_j(t-i)$ is the j-th component of $\boldsymbol{s}(t-i)$ and $\hat{\mu}$ is the baseline firing rate, which we can estimate by the mean of the response, $\sum_{t=1}^{T} r(t)/T$. We denote as $\boldsymbol{\beta}$ the vector of coefficients $(\beta_{11}, ..., \beta_{1d}, ..., \beta_{p1}, ..., \beta_{pd})$.

The nonlinear function $f(\cdot) : \mathbb{R} \to \mathbb{R}$ is defined as a linear combination of a fixed set of basis functions $f_k(\cdot) : \mathbb{R} \to \mathbb{R}$, $k \in \{1, ..., q\}$. Each basis function $f_k(\cdot)$ has a single input. Such functions are defined as piecewise linear functions determined by a predefined set of equidistant nodes $\{\delta_1, ..., \delta_q\}$ that covers the entire range of the stimulus. Figure 1(a) shows a representation of these functions for $q = 10$.

Therefore, given a second vector of coefficients $\boldsymbol{\alpha} = (\alpha_1, ..., \alpha_q)$ and some univariate stimulus s_j, $f(\cdot)$ is defined as

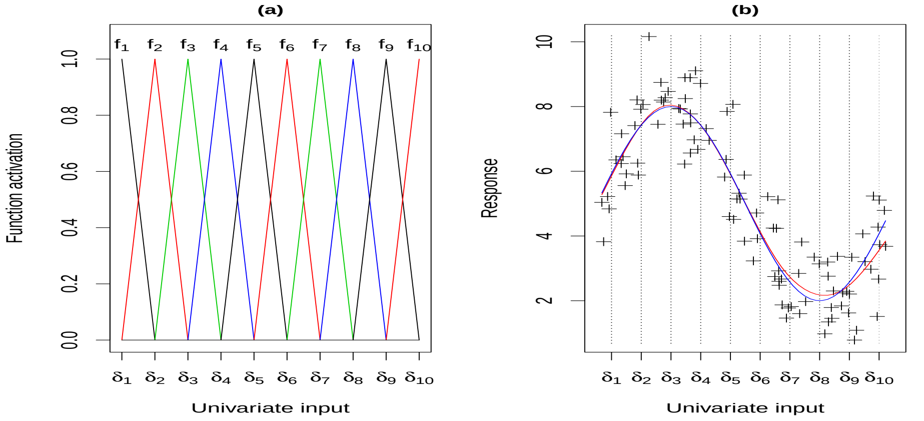

Fig. 1. (a) Piecewise linear functions used by the bilinear and fullrank models ($q = 10$). (b) A cubic smoothing spline fit to a univariate input, with knots $\{\delta_1, ..., \delta_{10}\}$. The red line is the fitting curve and the blue line is the true function.

$$f(s_j) = \sum_{k=1}^{q} \alpha_k f_k(s_j), \tag{5}$$

where

$$f_k(s_j) = \begin{cases} (s_j - \delta_{k-1})/(\delta_k - \delta_{k-1}) & \text{if } k > 1 \text{ and } \delta_{k-1} \leq s_j < \delta_k \\ (\delta_{k+1} - s_j)/(\delta_{k+1} - \delta_k) & \text{if } k < q \text{ and } \delta_k \leq s_j < \delta_{k+1} \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

Thus, vectors $\boldsymbol{\beta}$ and $\boldsymbol{\alpha}$ need to be estimated, resulting in a total of $pd + q$ parameters. This is done by updating $\boldsymbol{\beta}$ by ordinary least squares for a fixed $\boldsymbol{\alpha}$ and then updating $\boldsymbol{\alpha}$ by ordinary least squares for a fixed $\boldsymbol{\beta}$. Starting from an arbitrary initial value for either $\boldsymbol{\beta}$ or $\boldsymbol{\alpha}$, this step is repeated until convergence. This is known as alternating least squares (Young et al, 1976).

As noted by Ahrens et al (2008), the bilinear model is related to separable receptive fields (DeAngelis et al, 1995). When the receptive field is separable, the response cannot be expressed as a product of a function that only depends on time and a function that only depends on space (here, on the basis expansion instead).

On the other hand, the *fullrank* model is determined by qpd instead of $q+pd$ parameters. The fullrank model is defined as:

$$\hat{r}(t) = \hat{\mu} + \sum_{i=1}^{p} \sum_{j=1}^{d} \sum_{k=1}^{q} \gamma_{ijk} f_k(s_j(t-i)), \tag{7}$$

where functions $f_k(\cdot) : \mathbb{R} \to \mathbb{R}$ are defined as before.

Parameters γ_{ijk} are estimated by least squares. In both the bilinear and the fullrank schemes, the parameters are estimated from a fixed train data set with

N response values and p stimuli values per response. These parameters will be used to estimate future responses.

Both the bilinear model and the fullrank model have three shortcomings. First, the basis functions are data-independently defined. Second, they do not consider interactions between input variables, so the nonlinear power of the models is limited. We discuss the basic details of nonlinear regression below. Third, the learnt parameters are not time-varying, so that the response function is always the same. A fixed noise standard deviation $\sigma(t) = \sigma$ is also assumed. In the following section, we propose the use of some nonlinear techniques to overcome these limitations.

3 Basis expansions and kernel smoothing methods

In this section, we survey and discuss how to apply some nonlinear approaches to spike train data to obtain better results. Whatever approach we follow, we must somehow control the nonlinearity or complexity of the model. More complex models are less biased in exchange for increased variance. In general terms, we would choose simpler models for limited or ill-posed data and more complex models for well-behaved data. The objective is a model that optimizes the bias-variance trade-off, that is, that minimizes the expected prediction error. Automatic, data-dependent methods are preferred to control the model complexity. For example, techniques based on regularization (Hoerl and Kennard, 1970; Tibshirani, 1996), are useful for adjusting the complexity of the model and restricting its variance by imposing some constraint on the model parameters. Regularization can also achieve other goals, like variable selection.

There are two fundamental approaches for achieving nonlinearity. First, we can seek a more complex model than the linear model by establishing a linear combination of some basis expansions of the original terms (Schumaker, 2007). This is the approach taken by the bilinear and fullrank models. Second, we can fit simple (linear) models for different areas of the data domain (Loader, 1999), accounting for time-varying subject states.

Regression splines (Schumaker, 2007) is a family of popular nonlinear models that makes use of basis expansions of the original terms. In the univariate case, the input domain is divided into contiguous intervals, separated by a fixed set of knots. Whereas the placement of the knots can be data-driven, the number of knots is often specified by the user. In each interval, a polynomial function of order M is fitted, in such a way that the entire function is continuous, and has continuous derivatives up to order $M-2$ to assure continuity. Figure 1(b) shows a univariate cubic regression spline fit ($M = 4$) with $q = 10$ knots.

A more flexible, likewise spline-based, model is known as smoothing splines. Here, a maximal set of knots is used, and complexity is controlled by penalizing (regularizing) the curvature of the fitted function. These models can be extended to the multivariate case, giving rise to very flexible models. In this paper, we make use of two different spline methods, with a different degree of flexibility (complexity).

The first is known as *multivariate adaptive regression splines* (MARS), devised by Friedman (1991), where the basis functions are linear splines. For simplicity, we assume a unidimensional stimulus ($d = 1$). For each input variable, indexed by $i \in \{1, ..., p\}$, and each instance in the training data set, indexed by n, two piecewise linear functions are defined, $f^1_{ni}(\cdot) : \mathbb{R}^p \to \mathbb{R}$ and $f^2_{ni}(\cdot) : \mathbb{R}^p \to \mathbb{R}$, with one knot at $s(n-i)$. For a given vector $\boldsymbol{x} \in \mathcal{R}^p$, these are defined as

$$f^1_{ni}(\boldsymbol{x}) = \begin{cases} x_i - s(n-i) & \text{if } x_i > s(n-i) \\ 0 & \text{otherwise} \end{cases} \quad f^2_{ni}(\boldsymbol{x}) = \begin{cases} s(n-i) - x_i & \text{if } x_i < s(n-i) \\ 0 & \text{otherwise} \end{cases}$$

The MARS algorithm performs a forward greedy search, sequentially adding terms $h_{i^*}(\cdot)$, which are a single piecewise linear function or a product of two (or more) piecewise linear functions. To prevent overfitting, a backward deletion procedure is applied afterwards. Generalized cross-validation (Craven and Wahba, 1979) is used to decide how much the model should be pruned. Thus, MARS considers interactions between the stimuli. The resulting model has the form

$$\hat{r}(t) = \hat{\mu} + \sum_{i^*=1}^{p^*} \beta_{i^*} h_{i^*}(s(t-i)^p_{i=1}), \tag{8}$$

where p^* is the number of terms included into the model, $s(t-i)^p_{i=1}$ represents a vector with the values of the stimulus from $t-1$ to $t-p$ and β_{i^*} are the parameters for each term, which are estimated by least squares. The maximum number of functional terms in such products can be considered as a parameter of the algorithm. MARS produces continuous models with continuous derivatives.

On the other hand, we consider the *sparse additive model* (SpAM), devised by Ravikumar et al (2009). SpAM employs regularization instead of greedy searching to control the flexibility of the model. Like the additive model, proposed by Hastie and Tibshirani (1999), SpAM considers an additive linear combination of univariate functions and ignores interactions among the input variables:

$$\hat{r}(t) = \hat{\mu} + \sum_{i=1}^{p} f_i(s(t-i)), \tag{9}$$

where each $f_i(\cdot) : \mathbb{R} \to \mathbb{R}$ is, e.g., a cubic regression spline (see Figure 1(b)).

To achieve sparseness, SpAM employs an L_1-penalty (Tibshirani, 1996) imposed on the component L_2-norms of the functions, given by $(\int_0^T f_i^2(t)dt)^{1/2}$, so that the magnitude of the functional predictors is penalized. This leads to a number of terms (depending on the value of some regularization parameter) being effectively discarded. Unlike the classic additive model, regularization allows SpAM to be used in high-dimensional settings, as it is more interpretable, less sensitive to overfitting and also computationally efficient. The estimator is obtained by formulating a convex optimization problem, which is solved with a backfitting algorithm, described in (Hastie and Tibshirani, 1999).

Note that, whereas we need to manually setup the basis expansion configuration for the bilinear and fullrank models beforehand, both MARS and SpAM can automatically adjust model complexity during the training process. In addition, MARS can account for interactions among the predictors.

On the other hand, kernel smoothing models fit a simple model at each query point. For example, linear local regression (Loader, 1999) fits a linear regression model for each query point (for each spike count to be predicted), using some weighted neighborhood composed of the closest data to the query item. Here we apply this idea to the nonlinear methods described above.

Data (within the neighborhood) are typically weighted according to some kernel function $K_\tau()$ and the distances to the target item. The distances are measured either on the input variable space or can be supplied by the domain itself. In our case, the time dimension perfectly suits for our purposes. Let $n \in \{t-N, ..., t-1\}$, so that the distances are computed as the number of time slices between the target item and each previous data item, so that data items closer in time will be given more importance than further data items. We compute the weights w_n with the well-known tricube kernel function:

$$w_n = K_\tau(t-n+1) = \begin{cases} (1-(t-n+1)^3)^3 & \text{if } (t-n+1) \leq \tau \\ 0 & \text{otherwise,} \end{cases} \tag{10}$$

where τ is a smoothing parameter that indicates the width of the neighborhood. We need to estimate τ, which can be unique for all predictions or adaptively selected. For simplicity, we consider a unique smoothing parameter, $\tau = N$, in this paper, so that all data items in the built-in data set are used. Remember that the neighborhood only includes past data items.

Note that this weighting scheme can be applied to any of the algorithms presented above (bilinear, fullrank, MARS and SpAM), just by computing the weights and weighting the data set accordingly. This would yield their corresponding local versions. For an efficient model assessment, generalized cross-validation (Hastie et al, 2008) is a very efficient approximation to leave-one-out cross-validation for Gaussian data.

In summary, this paper investigates the effect of different types of nonlinearity to improve spike firing rate prediction for averaged trials. Firstly, the use of regression splines methods (MARS and SpAM) aims to deal the high complexity of neurological processing. Besides, we consider combinations of different stimuli instead of pure additive models by using MARS. Secondly, kernel smoothing can incorporate the subject evolution into the model and suppress the assumption of a stationary model by using only the recent subject states to build the model. Also, it avoids the fixed noise variance assumption. The bilinear and fullrank models, as suggested by Ahrens et al (2008), use the same model (obtained from a separate training data set) to predict all future data items, thus ignoring the internal evolution of the subject. The methods to be tested are:

	Non spline-based	Spline-based
Stationary	Bilinear, fullrank	MARS, SpAM
Local	Local bilinear, local fullrank	Local MARS, local SpAM

4 Experiments on synthetic data

We first study the different nonlinear approaches on five different synthetic scenarios or spike generating processes, following a similar experiment design than Ahrens et al.'s (2008). The first three scenarios are the same as the three data models used by Ahrens et al (2008). The remaining two are proposed here so as to reflect dynamic changes in the subject stimulus-response function. Results are evaluated by their predictive power (Sahani and Linden, 2003).

In all experiments we generate a one-dimensional stimuli vector, $s(t)_{t=1}^{T}$, from a normal distribution $\mathcal{N}(0,1)$, with $T = 1000$ time points. From this stimuli vector we obtain a firing probability $P(t)_{t=1}^{T}$ for each scenario according to one of the five generating processes described below (plus some uniform noise, $\mathcal{U}(-0.1, 0.1)$). All generating processes take into account the last $\eta = 10$ time points (the previous η stimuli). The first η values of $P(t)_{t=1}^{T}$ are set to zero by convention. Each resulting vector $P(t)_{t=1}^{T}$ is scaled to lie in $[0,1]$.

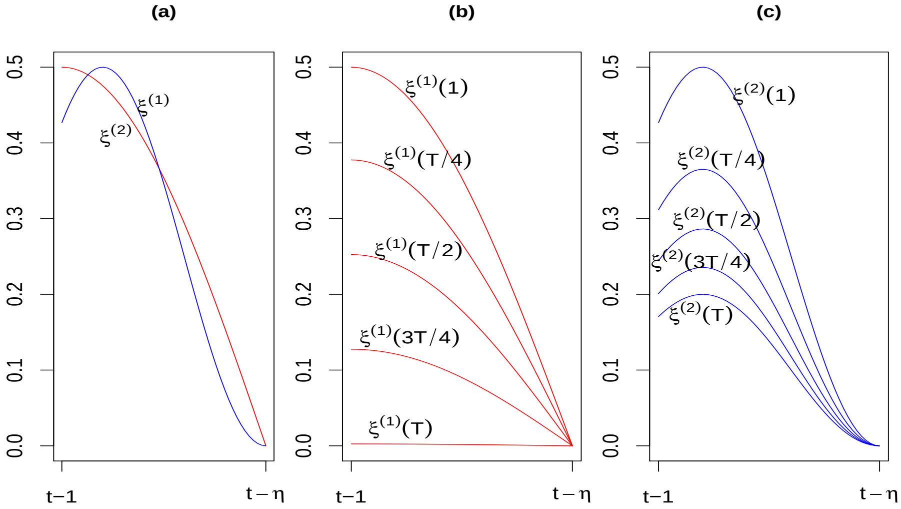

Fig. 2. (a) Stationary filter vectors $\boldsymbol{\xi}^{(1)}$ and $\boldsymbol{\xi}^{(2)}$. (b) Non-stationary filter vector $\boldsymbol{\xi}^{(1)}(t)$ at various time points. (c) Non-stationary filter vector $\boldsymbol{\xi}^{(2)}(t)$ at various time points.

The five generating processes are:

- **I. One stationary filter**. $P(t) = \sum_{i=1}^{\eta} \xi_i^{(1)} (s(t-i))^2$, where $\xi_i^{(1)} = sin(v_i)/2$ and $\boldsymbol{v}$ is a vector containing η equidistant increasing values (in radians) between $\pi/2$ and π.
- **II. Two stationary filters**. $P(t) = \sum_{i=1}^{\eta} \xi_i^{(1)} s(t-i) + \sum_{i=1}^{\eta} \xi_i^{(2)} (s(t-i))^2$, where elements $\xi_i^{(1)}$ are defined as in process I, and $\xi_i^{(2)} = (sin(v_i)+1)/4$. Here, $\boldsymbol{v}$ is a vector containing η equidistant increasing values (in radians) between $\pi/4$ and $3\pi/2$. Figure 2(a) shows filter vectors $\boldsymbol{\xi}^{(1)}$ and $\boldsymbol{\xi}^{(2)}$.

– **III. Nonlinear feature selective process**. $P(t) = \sum_{i=1}^{\eta} \mathcal{I}(|s(t-i) - v_i| < 0.2(max(s(t)_{t=1}^{T}) - min(s(t)_{t=1}^{T}))$, where $\mathcal{I}(\cdot)$ equals 1 when its argument is true and 0 otherwise, and $\boldsymbol{v}$ is a vector containing η equidistant increasing values between $min(s(t)_{t=1}^{T})$ and $max(s(t)_{t=1}^{T})$.
– **IV. One non-stationary filter**. $P(t) = \sum_{i=1}^{\eta} \xi_i^{(1)}(t)(s(t-i))^2$, where $\xi_i^{(1)}(t) = (T - t + 1)sin(v_i)/2T$, and $\boldsymbol{v}$ is a vector containing η equidistant increasing values (in radians) between $\pi/2$ and π. Figure 2(b) shows the non-stationary filter vector $\boldsymbol{\xi}^{(1)}(t)$ for several time points.
– **V. Two non-stationary filters**. $P(t) = \sum_{i=1}^{\eta} \xi_i^{(1)}(t)s(t-i) + \sum_{i=1}^{\eta} \xi_i^{(2)}(t)(s(t-i))^2$, where elements $\xi_i^{(1)}(t)$ are defined as in process IV, and $\xi_i^{(2)}(t) = (sin(v_i)+1)/u_t$. Here, $\boldsymbol{v}$ is a vector containing η equidistant increasing values (in radians) between $\pi/4$ and $3\pi/2$ and $\boldsymbol{u}$ is a vector containing T equidistant increasing values in the interval $[4, 10]$. Figure 2(c) shows the non-stationary filter vector $\boldsymbol{\xi}^{(2)}(t)$ for several time points.

At each time bin, the generating processes can produce up to twelve spikes. Hence, for each generating process, we obtain five spike trains of length T from a binomial distribution $Binom(12, P(t))$, $t = 1, ..., T$. By averaging such spike trains over the five trials, we compute the observed firing rate $r(t)_{t=1}^{T}$.

We have tested the bilinear and fullrank models, MARS and SpAM, and their local counterparts (L.bilinear, L.fullrank, L.MARS and L.SpAM) on these five experimental settings. For each generating process, we sampled ten pairs $(P(t)_{t=1}^{T}, r(t)_{t=1}^{T})$, corresponding to ten simulated neurons, from the same stimuli vector $s(t)_{t=1}^{T}$. All models have been trained taking $p = 20$ previous stimuli into account. For the non-local methods, we built the models on the first half of the data, using $N = T/2 - p$ data items, and we tested them on the second half. For the local methods, a different model is built for each data item in the second half of the data, where in the training data set includes the $N = 300$ last responses. The bilinear and the fullrank models were trained using $q = 5$ piecewise functions.

An advantage of the bilinear and fullrank methods is that these models can be graphically expressed, providing a compact description of the neuronal function. For instance, Ahrens et al (2008) show several plots. The MARS and SpAM models, since they are non-parametric, are more complex to be graphically described. However, one can still use simple graphs to ascertain the importance of the predictors within the model. Figure 3 (top graphs) shows measures of the importance of each predictor for MARS (left) and SpAM (right), for data obtained from generating process II. Each line represents one neuron. In the MARS case, this is the highest absolute coefficient β_{i^*} (Equation (8)) that involves each predictor. In the SpAM case, this is the L_2-norm of functions $f_i(\cdot)$ (Equation (9)) of each predictor. The eight bottom graphs of Figure 3 depict the univariate functions for the eight most relevant predictors in the SPaM models, giving an idea of the influence of these predictors on the response. Again, each line represents one neuron.

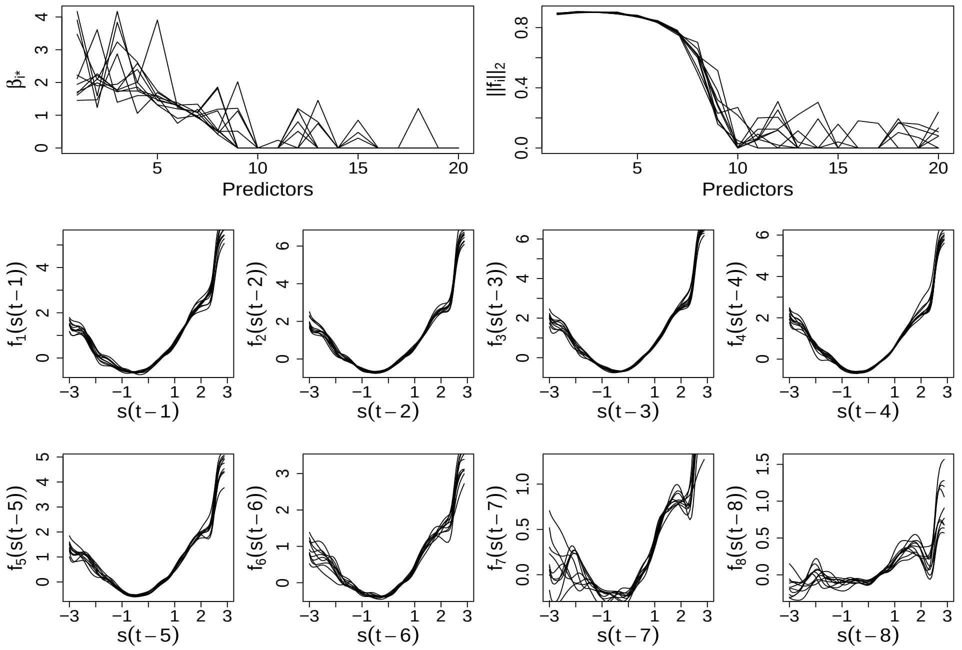

Fig. 3. Predictor importance for the stationary MARS (top left) and stationary SpAM (top right) models, and the eight most relevant functional predictors in the stationary SpAM models, for generating process II.

Figure 4 and Figure 5 illustrate the model fits for the local bilinear model and local SpAM, respectively, at various time points ($t = 501, 667, 834, 1000$) for generating process I. Within each figure, each pair of graphs is related to a time point. Each line represents a different neuron. Local bilinear models are represented by coefficients β_{i1} of Equation (4) (left graphs) and coefficients α_k of Equation (5) (right graphs). Local SpAM models are represented by the L_2-norm of functions $f_i(\cdot)$. Figure 6 and Figure 7 show the same for generating process IV.

Note that the models that correspond to generating process I (stationary) vary less across the different time points ($t = 501, 667, 834, 1000$) than those of generating process IV (non-stationary). This indicates to the practitioner that, for generating process I, a global model is preferable, whereas, for generating process IV, a local model is more adequate. Valuable biological insight about the underlying biological process can be extracted from this fact.

Figures 8-12 depict the true firing rate (blue) superimposed on the estimated firing rate (red), for some simulated neuron (the same for all figures) and $t = 901, ..., 1000$. Focusing on the stationary models (left graphs), simple visual inspection appears to indicate that the fullrank and bilinear models predict the firing rate approximately as well as the more complex MARS and SpAM models for all generating processes, although the performance of the bilinear model is

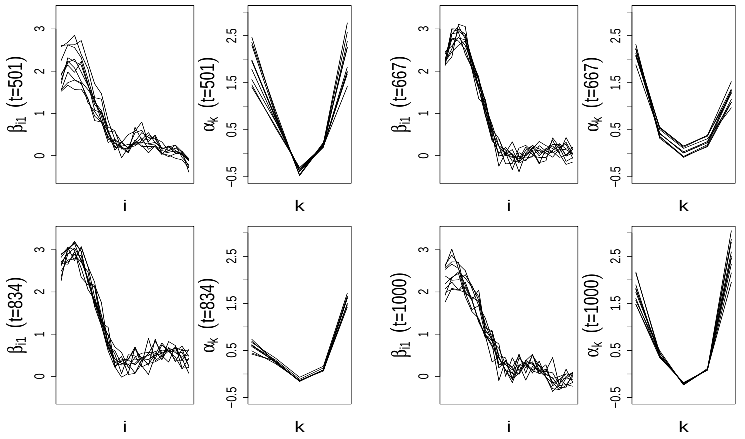

Fig. 4. Local bilinear models for generating process I at various time points.

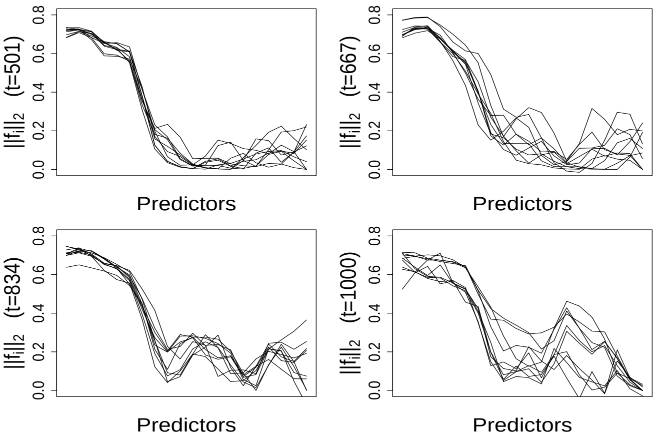

Fig. 5. Predictor norms of the local SpAM fits for generating process I at various time points.

slightly worse for generating process III, whose nonlinearity is more difficult to capture. In the local models (right graphs), however, L.SpAM seems to do the best job overall. L.SpAM is more accurate than L.MARS because L.MARS considers interactions between the inputs, whereas L.SpAM focuses the nonlinear strength separately at each input. Since the local models are built with fewer training data items (lower N) than the stationary models and there is no input interaction in any of the generating processes, MARS may slightly overfit the data. As expected, all local models behave much better for generating processes IV and V because they are non-stationary.

Figure 13 illustrates a quantitative comparison. It reports the predictive power (Sahani and Linden, 2003) of the methods, comparing each non-local method with its local counterpart. Each point represents a neuron simulated

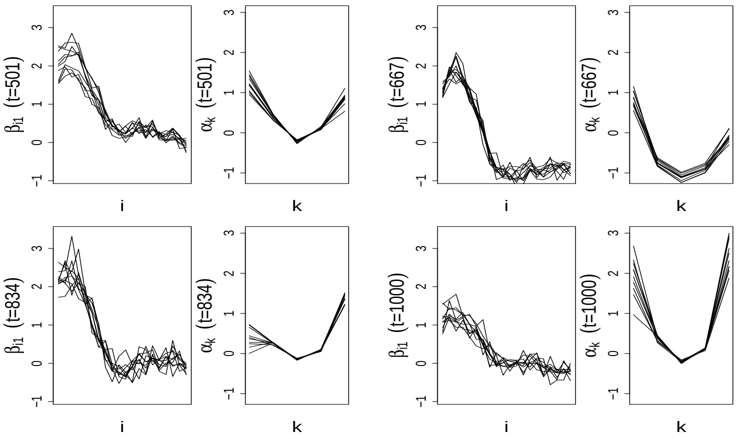

Fig. 6. Local bilinear models for generating process IV at various time points.

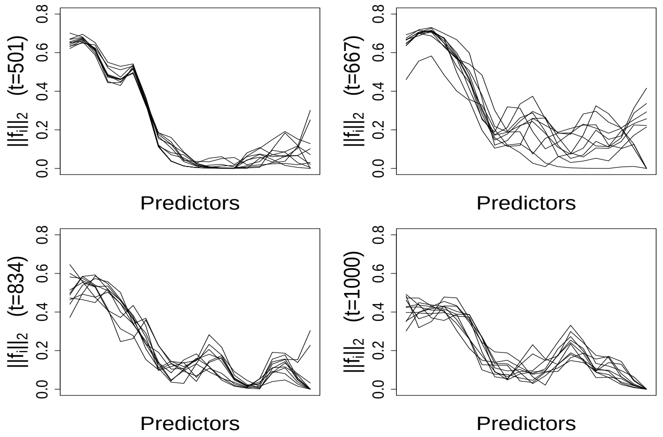

Fig. 7. Predictor norms of the local SpAM fits for generating process IV at various time points.

by each of the five generating processes. Generating processes are distinguished by different kinds of symbols. In the left-hand graphs, if one point lies on the left side of the dashed (diagonal) line, then the predictive power of the local method is greater than that of the non-local method, and the opposite applies if the point lies on the right side. The right-hand graphs (and horizontal histograms) show the predictive power difference between each local and non-local method as a function of the noise power (Sahani and Linden, 2003). It is clear that the local methods excel for generating processes IV and V, whereas the non-local methods are better for generating processes I, II and III. The biggest difference is for generating process V, where the local methods are much better than the non-local methods.

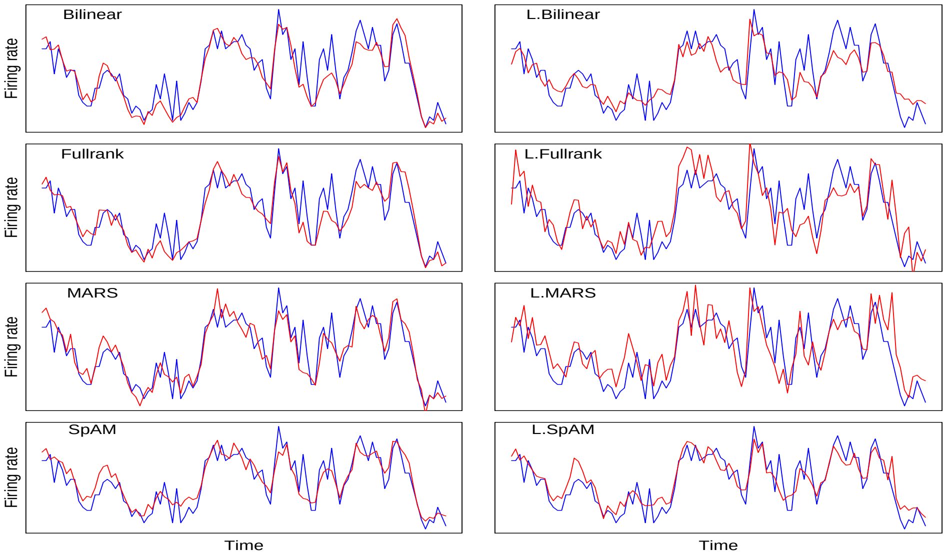

Fig. 8. Predicted signal (red) and true firing rate (blue) for generating process I in some time range for one neuron.

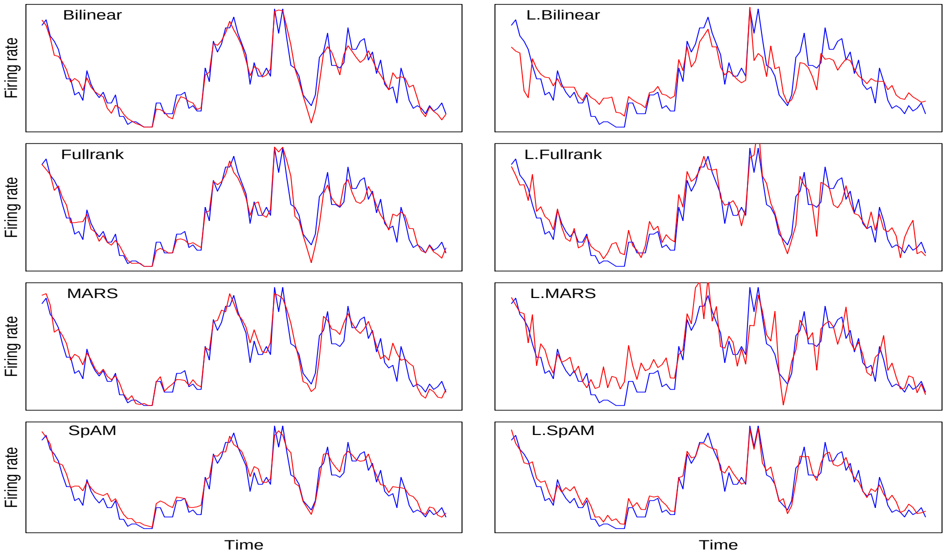

Fig. 9. Predicted signal (red) and true firing rate (blue) for generating process II in some time range for one neuron.

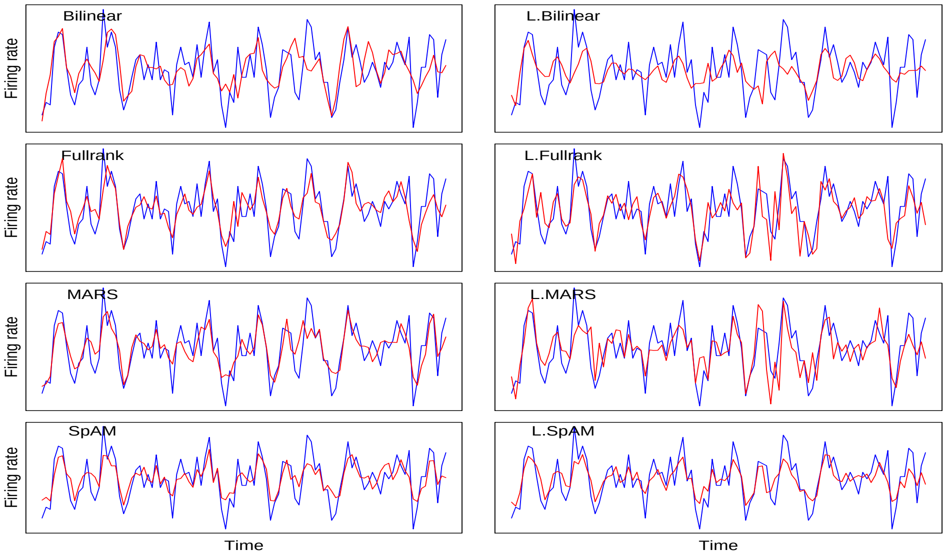

Fig. 10. Predicted signal (red) and true firing rate (blue) for generating process III in some time range for one neuron.

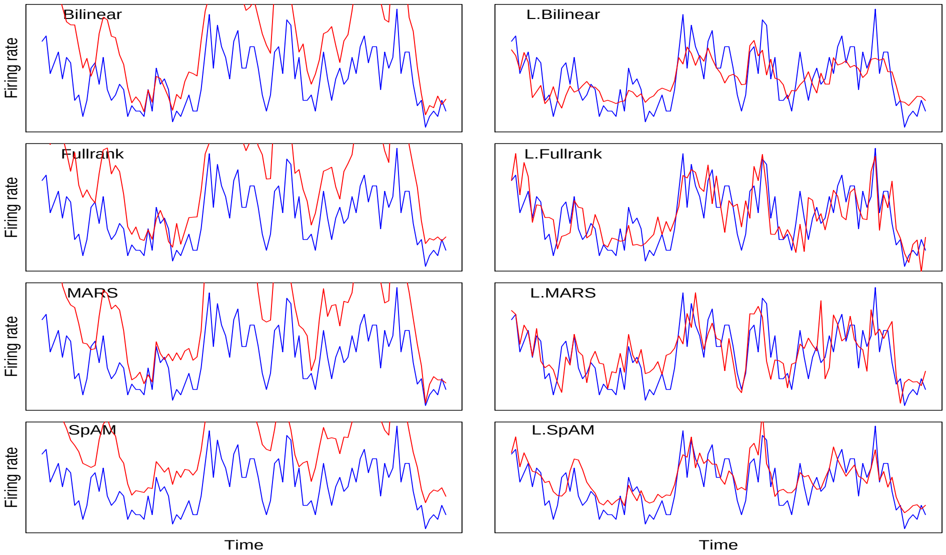

Fig. 11. Predicted signal (red) and true firing rate (blue) for generating process IV in some time range for one neuron.

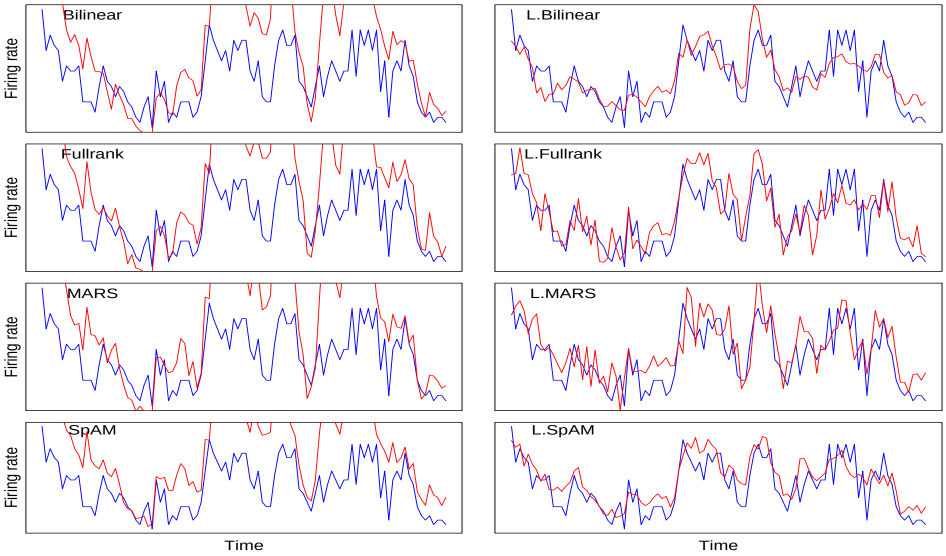

Fig. 12. Predicted signal (red) and true firing rate (blue) for generating process V in some time range for one neuron.

Figure 14 illustrates a comparison of L.bilinear and L.fullrank against L.MARS and L.SpAM. Interestingly, it now is clear that L.SpAM outperforms both L.bilinear and L.fullrank for all generating processes. L.MARS, however, only behaves much better for the generating process III, which is a nonlinear process that the bilinear and fullrank methods cannot entirely capture.

From this experiment, we can conclude that, if there is no interaction between the stimuli, SpAM is a handy algorithm for a broad spectrum of neural spike estimation scenarios, since regularization can automatically adjust the complexity of the model. Interactions between the stimuli are studied in the next section. Also, locality should be taken into account when the encoding function is suspected to vary over time. We believe that the use of stationary models to characterize neuron responses can sometimes lead to inaccurate predictions and is based on an often unrealistic assumption. In the next section, we check these hypotheses on a real scenario.

5 Experiments on real data

We have also investigated the impact of nonlinearity on real data, in particular large-scale neuronal recordings in cat primary visual cortex (area 17). The data were collected by Tim Blanche at the laboratory of Nicholas Swindale, University of British Columbia, and can be downloaded from the NSF-funded CRCNS Data Sharing website[1].

[1] http://crcns.org

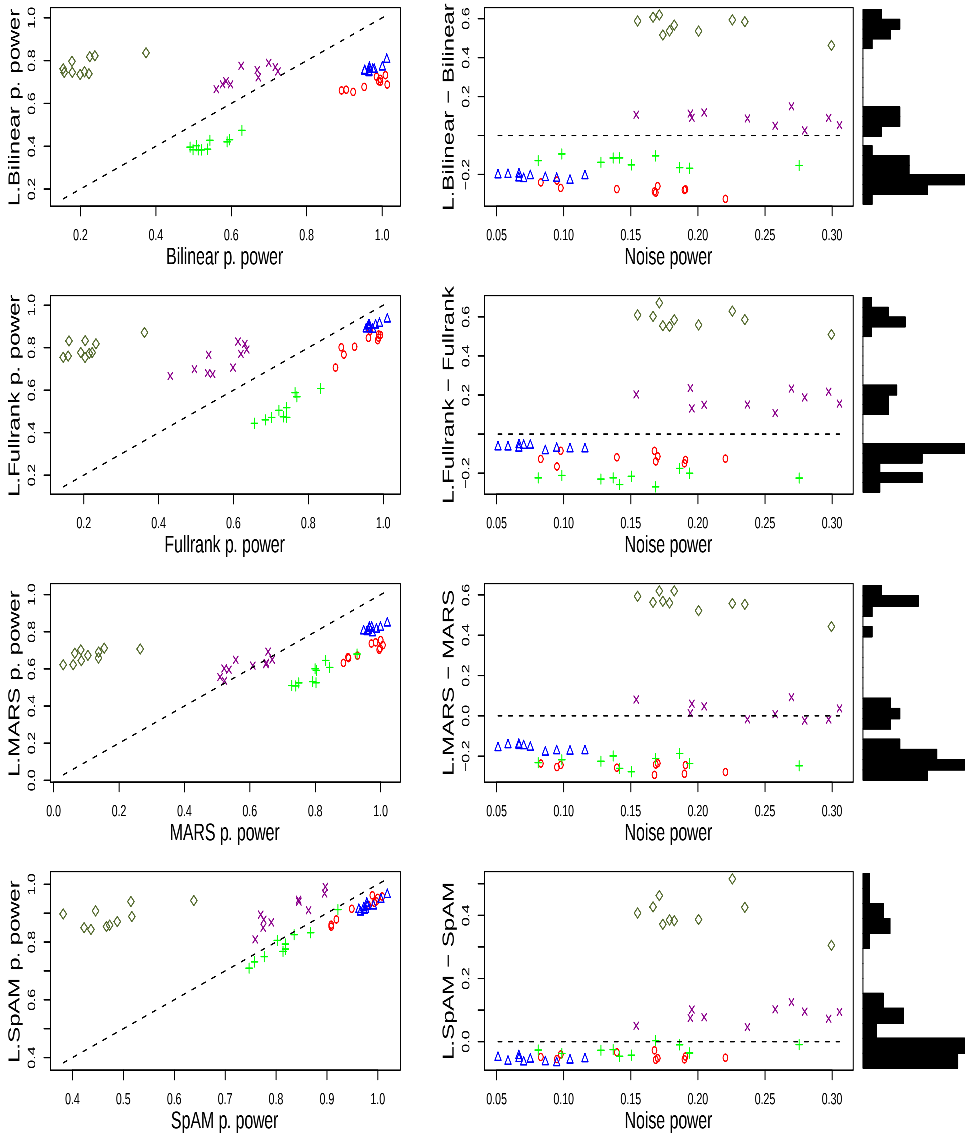

Fig. 13. Comparison of the non-local methods against their local counterparts in terms of predictive power (p.power). Red ○-dots correspond to generating process I, blue △-dots correspond to generating process II, green +-dots correspond to generating process III, magenta ×-dots correspond to generating process IV and grey ◇-dots correspond to generating process V.

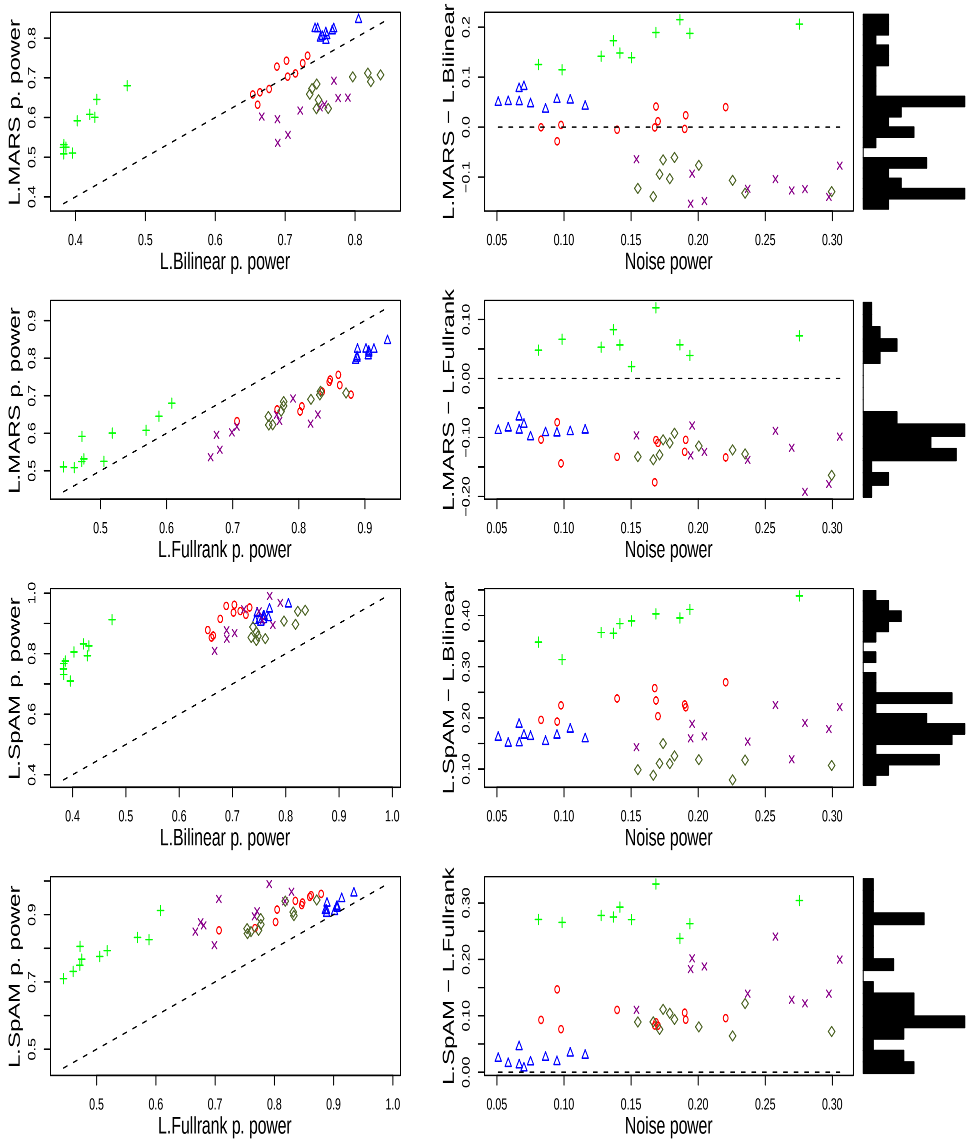

Fig. 14. Comparison of the local bilinear and fullrank models against the local MARS and local SpAM methods in terms of predictive power (p.power). Red ○-dots correspond to generating process I, blue △-dots correspond to generating process II, green +-dots correspond to generating process III, magenta ×-dots correspond to generating process IV and grey ◇-dots correspond to generating process V.

Data corresponds to extracellular neural activity under several types of visual stimuli. We work with the simplest kind of stimulus, consisting of an oriented drifting bar moving on a screen. The drifting bar moves in 18 different directions.

The data set contains eight trials of spiking data for ten (simultaneously recorded) neurons. At each trial, the 18 stimulus values are presented in random order for approximately 4 seconds each. We have partitioned the time range in bins of 100ms, counting the number of spikes at each bin. Therefore, there are 40 bins per stimulus value and $T = 720$ time bins in total.

Note that encoding the stimulus as the number of degrees or as a categorical variable is an incoherent representation. For example, if we represent the bar orientation as the number of degrees, we are implying that the 0° orientation lies far away from the 340° orientation, whereas they are actually only 20° apart. Instead, we use two variables to represent each orientation, $s_1(t) = 0.5\, sin(radians(t))$ and $s_2(t) = 0.5\, cos(radians(t))$. These pairs are Cartesian coordinates on the circumference of diameter 1.0. In this way, we have the maximum Euclidean distance (1.0) between "opposite" stimuli.

Note also that we cannot directly average the spike counts across the trials, because the stimulus values are presented in a different order at each trial. Noise cannot be assumed to be Gaussian (Equation (3)) if trials are not averaged and, hence, the aforementioned methods cannot be applied. Instead, we reorder each trial's spike counts so that the 40 bins of the 0° orientation are followed by the 40 bins of the 20° orientation, and so on. We now average the spike counts across the reordered trials. Figure 15 illustrates the (averaged) firing rate for two neurons. The graphs on the left are not ordered, so each point in the series is the mean of the spike counts at the same time point. The graphs on the right are ordered, so each point in the series is the mean of spike counts at different time points within the trials (with the same stimulus value, however).

It is obvious that we cannot use previous stimuli for firing rate prediction if they correspond to different stimulus values. However, we hypothetize that the response does not depend here on previous stimulus values, but on the current stimulus value and the amount of time it has been held.

In Figure 15 (left graphs), the "non-reordered" firing rate does not appear to follow a clear pattern. Firstly, the smoothed firing rate is less informative for the non-reordered version. For example, the red curve is very flat for the neuron on top, indicating that there is no substantial change across the entire time range. Interestingly, the right graphs are more informative. The neurons are clearly more active for certain ranges of the stimulus. For instance, the neuron on top fires mostly for stimulus values around 80°-140° and stimulus values around 260°-320°, which are opposite orientations and same direction. Secondly, if we observe the high-frequency scale, it is patent that, in the reordered spike counts, the spiking pattern within each segment of 40 bins is roughly repeated in neighboring segments (segments concerning close stimulus values). For instance, the pattern highlighted in blue in Figure 15 (right, bottom graph) is almost identical, up to some scaling factor, to that in the neighboring segments. Although

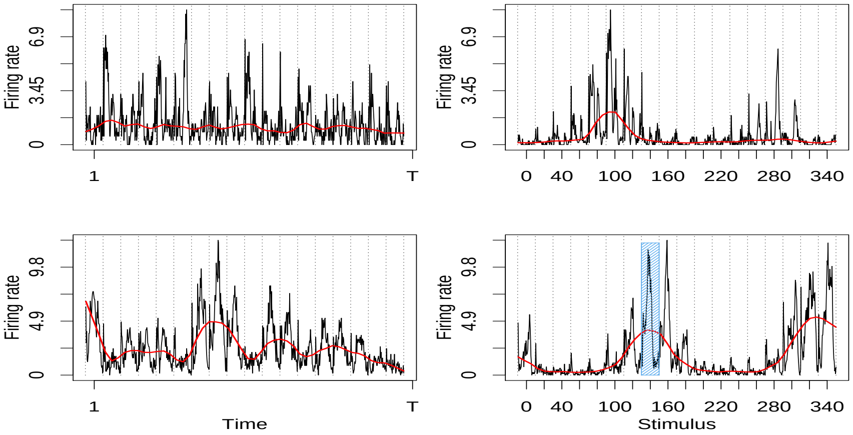

Fig. 15. Mean spike counts for one neuron (top) and another neuron (bottom) before ordering (left) and after ordering the bins (right). Vertical dotted lines mark the change of stimulus. The red curves are a smoothed version of the spike counts. A common pattern is highlighted in blue.

not shown, the same applies for the other eight neurons. This encourages us to consider each 40-bin segment separately from preceding segments.

The firing rate $r(t)$ can thus be modeled as

$$\begin{aligned} r(t) &= g(\boldsymbol{z}(t-1)) + \epsilon(t) \qquad (11)\\ \boldsymbol{z}(t-1) &= (s_1(t-1), s_2(t-1), \theta_{\boldsymbol{s}}(t-1)), \end{aligned}$$

where $\theta_{\boldsymbol{s}}(t-1)$ is the number of time points (up to $t-1$) holding the same stimulus value $(s_1(t-1), s_2(t-1))$. We scale $\theta_{\boldsymbol{s}}(t-1)$ so that it lies into the interval $(-1, 1)$ (like $s_1(t-1)$ and $s_2(t-1)$).

Locality is now applied on the input space rather than on the time dimension. The weights w_n, defined in Equation (10), are now defined as

$$w_n = K_\tau(||\boldsymbol{z}(t) - \boldsymbol{z}(n)||_2) = \begin{cases} (1 - ||\boldsymbol{z}(t) - \boldsymbol{z}(n)||_2^3)^3 & \text{if } ||\boldsymbol{z}(t) - \boldsymbol{z}(n)||_2 \leq \tau \\ 0 & \text{otherwise,} \end{cases} \qquad (12)$$

where τ is the bandwidth parameter and $|| \cdot ||_2$ is the L_2-norm operator.

The bilinear and fullrank models, MARS, SpAM, and their local versions can be applied to a data set built according to Equation (11). However, stationary models are difficult to use here. If we train a model on a fixed part of the averaged spike counts data, only a subset of the possible stimulus values is used to build the model. Since none of the stimulus values are the same in the testing part, extrapolation is unlikely to work, and the prediction will be highly unstable. Therefore, we only consider here non-stationary (local) models. We have used the $N = 120$ closest responses (in the stimulus space) to build the models. Spikes have been estimated for $t = N + 1, ..., T$.

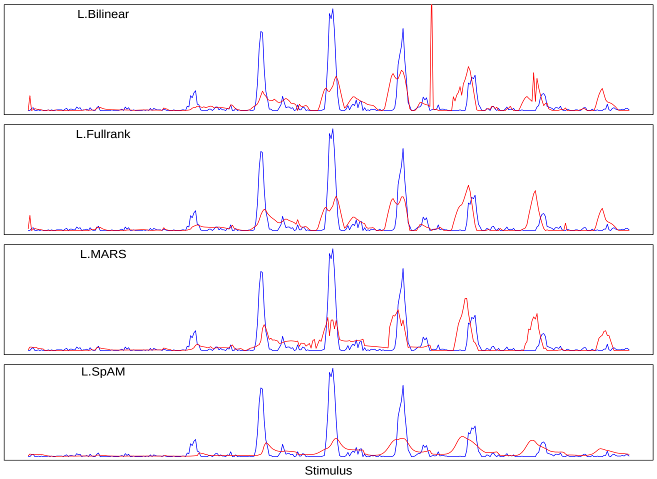

Fig. 16. Predicted signal (red) and true signal (blue) for neuron t18 from primary visual cortex (area 17) data.

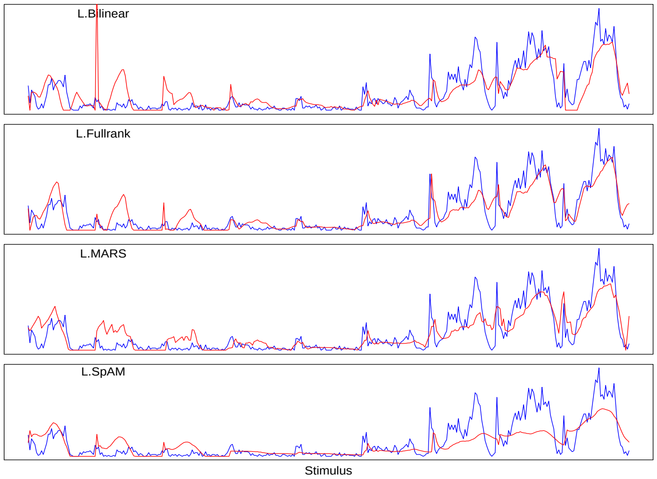

Fig. 17. Predicted signal (red) and true signal (blue) for neuron t27 from primary visual cortex (area 17) data.

Figure 16 and Figure 17 illustrate the estimated spike counts against the true estimated spike counts for two neurons, t18 and t27, respectively. It appears that L.MARS and L.fullrank models offer the best fit, whereas L.bilinear is slightly worse and L.SpAM (run with minimal regularization) is smoother than the others. L.MARS probably performs better than L.SpAM because it takes into account interactions among the inputs. The mean and standard deviation of the

Table 1. Quantiles of the p-values obtained from the Kolmogorov-Smirnov test based on the time-rescaling theorem.

Quantile	L.Bilinear	L.Fullrank	L.MARS	L.SpAM
0%	1.11e-15	2.22e-16	1.11e-15	2.22e-16
25%	2.22e-07	3.46e-08	2.80e-07	4.11e-08
50%	0.145	0.071	0.167	0.104
75%	0.789	0.758	0.793	0.789
100%	0.819	0.800	0.889	0.816

predictive power for the L.bilinear and L.fullrank models, L.MARS and L.SpAM are, respectively, 0.08(±0.41), 0.11(±0.26), 0.16(±0.43) and 0.06(±0.22).

To evaluate the models, we use the Kolmogorov-Smirnov test based on the time-rescaling theorem (Brown et al, 2001). In short, we compute rescaled times

$$v_a = 1 - exp\left[-\int_{u_a}^{u_{a+1}} \hat{r}(t)dt\right], \tag{13}$$

where $u_1 < ... < u_a < ... < u_A$ denote the set of individual spike times. It can be shown that the v_a values are independent uniformly distributed random variables if and only if the estimated response $\hat{r}(t)$ corresponds to the true conditional distribution of the process. Hence, to perform a usual Kolmogorov-Smirnov test, we just need to order the v_a values from the smallest one to the largest one and check if they are uniformly distributed.

Table 1 shows quantiles for the p-values obtained from applying the test over each model, each trial and each neuron. Hence, there are 8 × 10 p-values per algorithm. Note that L.MARS obtains the highest p-values, which reveals better fits. As observed, there are some cases where the estimated response is unlikely to be correct (low p-values). There are specific neurons in the experiment, indeed, that appear to be difficult to be modeled with the proposed models. We feel that a model including the activity of related neurons (Truccolo et al, 2005) or spike history terms (Paninski, 2004; Paninski et al, 2004; Truccolo et al, 2005) could be more adequate in these cases. There are other neurons, like t18 and t27, that are well modeled with these local models.

6 Discussion

In this paper, we have studied several nonlinear models on a number of different cases of spike firing rate prediction. Although all spike generating processes are intrinsically nonlinear in the synthetic scenarios, the source of their nonlinearity is certainly different. Whereas the first two of the above generating processes can be approximated by a simple extension of the linear model, more complex models are required to describe the third generating process. The fourth and fifth generating processes intend to simulate a response whose underlying model varies across the time. This could account for the habituation of the subject to the stimulus or any other internal change in the subject. In this case, local models that take into account this variation, even if they are relatively simple, definitely outperform other more complex stationary models.

Due to the huge variability of neural processes, it is impossible to choose a level of complexity, a kind of nonlinear approach or a family of models that universally fit well for the neural firing rate prediction problem. Some preliminary analysis is needed to ascertain the best model for a specific problem. For example, we studied the response of some neurons in the cat primary visual cortex (area 17) to simple stimuli. The models presented above had to be refined somewhat to tackle this problem. We finally found that the response of these neurons is to some extent independent from previous stimuli, the current stimulus value and the exposure time being the key inputs. In addition, the subject follows different patterns of response for different stimulus values. These patterns, however, are alike for close stimulus values. For this reason, locality plays a fundamental role in the prediction.

Of the studied models, bilinear, fullrank and SpAM are additive, i.e., they do not consider interactions among different stimuli. Therefore, they appear to deal with nonlinearity only at the earliest (dendritic) stages of neuron processing. Other models that apply nonlinearity on the output of a linear model intend to capture the processes at the latest stage of the neural process (spike generation). Models in the literature typically follow one of these approaches. However, to fully understand the encoding properties of a neuron, it may be necessary to consider interactions among different stimuli. We believe that they could be the basis of the intermediate processing stages of the neuron. MARS does consider interactions among stimuli and has output the best results in the real data experiments.

There exist other models in the statistics field to deal with interactions among the inputs. For example, the Component Selection and Smoothing Operator (COSSO) (Lin and Zhang, 2006) is a method based on L_1-regularization for simultaneous function selection and smoothing. COSSO is defined in the context of smoothing spline ANOVA (Wahba, 1990), where the estimated function potentially includes interactions of any order among the inputs. However, we have found that COSSO does not perform well for the neural spike count prediction task. Unlike MARS and SpAM, COSSO is not designed for high dimensional problems. Although smoothing spline ANOVA can capture more complex relations, it tends, in this particular case, to overfit in spite of regularization, and the resulting models are poorer than for MARS and SpAM.

Finally, note that local bilinear and local fullrank models can be adapted to single trials, following the guidelines introduced by Ahrens et al (2008). The extension of MARS and SpAM to this setting is a more complex issue.

Acknowledgments

This research was partially supported by the Spanish Ministry of Science and Innovation, projects TIN2010-20900-C04-04, Consolider Ingenio 2010-CSD2007-00018 and Cajal Blue Brain. We thank Pradeep Ravikumar and John Lafferty for providing us the SpAM code.

Bibliography

Ahrens MB, Paninski L, Sahani M (2008) Inferring input nonlinearities in neural encoding models. Network: Computation in Neural Systems 19:35–67

Averbeck BB, Sohn JW, Lee D (2006) Activity in prefrontal cortex during dynamic selection of action sequences. Nature Neuroscience 9:276–282

Bezdudnaya T, Cano M, Bereshpolova Y, Stoelzel CR, Alonso JM, Swadlow HA (2006) Thalamic burst mode and inattention in the awake LGND. Neuron 49:421–432

Brenner N, Bialek W, de Ruyter van Steveninck R (2000) Adaptive rescaling maximizes information transmission. Neuron 26:695–702

Brown EN, Barbieri R, Ventura V, Kass RE, Frank LM (2001) The time-rescaling theorem and its application to neural spike train data analysis. Neural Computation 14:325–346

Brown EN, Kass R, Mitra PP (2004) Multiple neural spike train data analysis: State-of-the-art and future challenges. Nature Neuroscience 7:456–461

Craven P, Wahba G (1979) Smoothing noise data with spline functions: Estimating the correct degree of smoothing by the method of generalized cross-validation. Numerische Mathematik 31:377–403

DeAngelis GC, Ohzawa I, Freeman RD (1995) Receptive-field dynamics in the central visual pathways. Trends of Neuroscience 18:451–458

Escola S, Fontanini A, Katz D, Paninski L (2011) Hidden Markov models for the stimulus-response relationships of multistate neural systems. Neural Computation 23:1071–1132

Friedman J (1991) Multivariate adaptive regression splines. Annals of Statistics 19:1–141

Gerstein GL, Perkel DH (1969) Simultaneously recorded trains of action potentials: Analysis and functional interpretation. Science 164:828–830

Haider B, Duque A, Hasenstaub AR, Yu Y, McCormick DA (2007) Enhancement of visual responsiveness by spontaneous local network activity in vivo. Journal of Neurophysiology 97:4186–4202

Hastie T, Tibshirani R (1999) Generalized Additive Models. Chapman and Hall

Hastie T, Tibshirani R, Friedman J (2008) The Elements of Statistical Learning: Data Mining, Inference and Prediction, 2nd edn. Springer

Hoerl A, Kennard R (1970) Ridge regression: Biased estimates for nonorthogonal problems. Technometrics 12:55–67

Kass RE, Ventura V, Brown EN (2005) Statistical issues in the analysis of neuronal data. Journal of Neurophysiology 94:8–25

Lin Y, Zhang HH (2006) Component selection and smoothing in multivariate nonparametric regression. Annals of Statistics 34:2272–2297

Loader C (1999) Local Regression and Likelihood. Springer

Machens CK, Wehr MS, Zador AM (2004) Linearity of cortical receptive fields measured with natural sounds. Journal of Neuroscience 24:1089–1100

Paninski L (2004) Maximum likelihood estimation of cascade point-process neural encoding models. Network: Computation in Neural Systems 15:243–262

Paninski L, Pillow JW, Simoncelli EP (2004) Maximum likelihood estimation of a stochastic integrate-and-fire neural encoding model. Neural Computation 16:2533–2561

Pillow JW, Simoncelli EP (2006) Dimensionality reduction in neural models: An information-theoretic generalization of spike-triggered average and covariance analysis. Journal of Vision 6:414–428

Ravikumar P, Lafferty J, Liu H, Wasserman L (2009) Sparse additive models. Journal of the Royal Statistical Society: Series B, 71:1009–1030

Sahani M, Linden JF (2003) How linear are auditory cortical responses? In: Becker S, Thrun S, Obermayer K (eds) Advances in Neural Information Processing Systems, 15, pp 125–132

Schumaker LL (2007) Spline Functions: Basic Theory. Cambridge Mathematical Library

Tibshirani R (1996) Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society: Series B, 58:267–288

Truccolo W, Eden UT, Fellows MR, Donoghue JP, Brown EN (2005) A point process framework for relating neural spiking activity to spiking history, neural ensemble and extrinsic covariate effects. Journal of Neurophysiology 93:1074–1089

Wahba G (1990) Spline Models for Observational Data. 59, SIAM

Young FW, de Leeuw J, Takane Y (1976) Regression with qualitative and quantitative variables: An alternating least squares method with optimal scaling features. Psychometrika 41:505–529