Online Analysis for Neural Spike Trains treated as a Time Series

David Carlson

May 17, 2013

1 Modeling

1.1 Basic Model

Let $\boldsymbol{x} \in \mathbb{R}^N$ be a time series. Assume we know the dictionary $\mathbf{A}$ of size K. We want to fit a sparse convolutional model to this dataset:

$$\begin{aligned}
\boldsymbol{x} &= \sum_{t=1}^{N} (z_t \delta_t) \star (\mathbf{A}\boldsymbol{y}_t) + \boldsymbol{\epsilon} && (1)\\
z_t &\sim \text{Bernoulli}(\pi) && (2)\\
\gamma_{t\,t=1}^{\;N} &\sim CRP(\alpha) && (3)\\
\boldsymbol{y}_t|\gamma_t = i &\sim \mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Lambda}_i^{-1}) && (4)\\
\boldsymbol{\epsilon} &\sim \text{AR}(1) && (5)\\
\boldsymbol{\mu}_i, \boldsymbol{\Lambda}_i &\sim \mathcal{NW}(\boldsymbol{\mu}_0, \kappa_0, \nu_0, \boldsymbol{\Phi}_0) && (6)
\end{aligned}$$

1.2 Extension to have an evolving mean

$$\begin{aligned}
\boldsymbol{x} &= \sum_{t=1}^{N} (z_t \delta_t) \star (\mathbf{A}\boldsymbol{y}_t) + \boldsymbol{\epsilon} && (7)\\
z_t &\sim \text{Bernoulli}(\pi) && (8)\\
z_t &\sim \text{Bernoulli}(\pi) && (9)\\
\gamma_{t\,t=1}^{\;N} &\sim CRP(\alpha) && (10)\\
\boldsymbol{y}_t|\gamma_t = i &\sim \mathcal{N}(\boldsymbol{\mu}_{it}, \boldsymbol{\Lambda}_i^{-1}) && (11)\\
\boldsymbol{\epsilon} &\sim \text{AR}(1) && (12)\\
\{\boldsymbol{\mu}_{i1}, ..., \boldsymbol{\mu}_{iN}, \boldsymbol{\Lambda}_i\} &\sim \prod_{k=1}^{K} \left[GP(\mu_{ik}; \mu_{0k}, \tau^{-1}\mathcal{K}(\cdot, \cdot|\beta)) \right] && (13)\\
& \quad \mathcal{W}(\boldsymbol{\Lambda}_i; \nu_0, \boldsymbol{\Phi}_0)] && (14)\\
\mathcal{K}(t_1, t_2|\beta) &= \exp(-\beta|t_1 - t_2|) && (15)
\end{aligned}$$

1.3 Extension to multi-channel models

Let $\mathbf{X} \in \mathbb{R}^{N \times C}$ be our data matrix, where C is the number of channels.

$$\begin{aligned}
\mathbf{X} &= [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_C] && (16)\\
\boldsymbol{x}_c &= \sum_{t=1}^{N} (z_t \delta_t) \star (\mathbf{A}\boldsymbol{y}_c t) + \boldsymbol{\epsilon}_c && (17)\\
z_t &\sim \text{Bernoulli}(\pi) && (18)\\
\boldsymbol{y}_{ct}|\gamma_t = i &\sim \mathcal{N}(\boldsymbol{\mu}_{ic}, \boldsymbol{\Lambda}_{ic}^{-1}) && (19)\\
\gamma_{t t=1}^{N} &\sim i \sim CRP(\alpha) && (20)\\
\boldsymbol{\epsilon}_c &\sim \text{AR}(1) && (21)\\
\{\boldsymbol{\mu}_i 1, \ldots, \boldsymbol{\mu}_{iC}, \boldsymbol{\Lambda}_i 1, \ldots, \boldsymbol{\Lambda}_{iC}\} &\sim H_0 && (22)\\
H_0 &= \prod_{c=1}^{C} [\prod_{k=1}^{K} \left[GP(\mu_{ick}; \mu_{c0k}, \tau_c^{-1}\mathcal{K}(\cdot, \cdot|\beta))\right] \\
&\quad \mathcal{W}(\boldsymbol{\Lambda}_i c; \nu_0, \boldsymbol{\Phi}_0)] && (23)\\
\mathcal{K}(t_1, t_2|\beta) &= \exp(-\beta|t_1 - t_2|) && (24)
\end{aligned}$$

2 Online Inference

We want to process the data in an online manner. First, we need to process the data in a windowed way, so let $\boldsymbol{x}_t^{-t}$ be P samples from $\boldsymbol{x}^{-t}$ starting at t where $\boldsymbol{x}^{-t} = \boldsymbol{x} - \sum_{n=1}^{n \neq t} (z_n \delta_n) \star (\mathbf{A}\boldsymbol{s}_n)$. For this window, we will have:

$$\begin{aligned}
\boldsymbol{x}_t &= z_t \mathbf{A} \boldsymbol{s}_t + \boldsymbol{\epsilon}_t && (25)\\
\boldsymbol{\epsilon}_t &\sim \mathcal{N}(0, \boldsymbol{\Sigma}) && (26)
\end{aligned}$$

To do this, we want to figure out whether there is a spike or not, so we need to infer the probability $p(z_t = 1|x_t)$:

$$\begin{aligned}
p(z_t = 1|\boldsymbol{x}_t) &= p(\boldsymbol{x}_t|z_t = 1)p(z_t = 1)/p(\boldsymbol{x}_t) && (27)\\
p(\boldsymbol{x}_t|z_t = 1) &= \sum_{i=1}^{C+1} p(\boldsymbol{x}_t|z_t = 1, \gamma_t = i)p(\gamma_t = i) && (28)\\
p(\boldsymbol{x}_t|z_t = 1, \gamma_t = i) &= \int p(\boldsymbol{x}_t|z_t = 1, \gamma_t = i, \boldsymbol{y}_t)p(\boldsymbol{y}_t|\gamma_t = i)d\boldsymbol{y}_t && (29)
\end{aligned}$$

We can use the CRP formulation to get the probabilities, so $p(\gamma_t = i) = \frac{n_i^{(t-1)}}{n_\cdot^{(t-1)} + \alpha}$ for a previously used cluster and $\boldsymbol{\phi}(\gamma_t = C + 1) = \frac{\alpha}{n_\cdot^{(t-1)} + \alpha}$ where C is the number of currently used clusters. We define $n_i^{(t-1)} = \sum_{n=1}^{t-1} z_t 1(\gamma_n = i)$. We need to use an approximation to get this quantity:

$$p(\boldsymbol{y}_t|\gamma_t = i) = \int \mathcal{N}(\boldsymbol{y}_t; \boldsymbol{\mu}_i, \boldsymbol{\Lambda}_i^{-1})\mathcal{N}\mathcal{W}(\boldsymbol{\mu}_i, \boldsymbol{\Lambda}_i; \hat{\boldsymbol{\mu}}_i, \kappa_i, \nu_i, \boldsymbol{\Phi}_i)d\boldsymbol{\mu}_i d\boldsymbol{\Lambda}_i \qquad (30)$$

If the precision matrix is reasonably known, then we can approximate this as:

$$p(\boldsymbol{y}_t|\gamma_t = i) \simeq \int \mathcal{N}(\boldsymbol{y}_t; \boldsymbol{\mu}_i, \boldsymbol{\Lambda}_i^{-1})\mathcal{N}(\boldsymbol{\mu}_i; \hat{\boldsymbol{\mu}}_i, (\kappa_i \boldsymbol{\Lambda}_i)^{-1})d\boldsymbol{\mu}_i \quad (31)$$

$$p(\boldsymbol{y}_t|\gamma_t = i) \simeq \mathcal{N}\left(\hat{\boldsymbol{\mu}}_i, (\frac{\kappa_i}{1+\kappa_i}\Lambda)^{-1}\right) \quad (32)$$

$$p(\boldsymbol{x}_t|z_t = 1, \gamma_t = i) \simeq \mathcal{N}\left(A\hat{\boldsymbol{\mu}}_i, \boldsymbol{\Sigma} + (\frac{\kappa_i}{1+\kappa_i}\Lambda)^{-1}\right) \quad (33)$$

At this point, we can approximately calculate $p(z_t = 1|\boldsymbol{x}_t)$. If $p(z_t = 1|\boldsymbol{x}_t) > .5$, then we would set z_t to 1. In order to make sure we're choosing the best spot to put the spike, I will choose the z_t with the highest $p(z_t|\boldsymbol{x}_t)$ in the range of samples of $[t, t+\tau]$ so that we can look slightly ahead. We choose γ_i to be the indicator with the highest a posteriori value.

The posterior for $\{\boldsymbol{\mu}_i, \boldsymbol{\Lambda}_i\}$ is not conjugate to $p(\boldsymbol{x}_t|\gamma_t = i, \boldsymbol{\mu}_i, \boldsymbol{\Lambda}_i)$. It is conjugate to $p(\boldsymbol{y}_t|\gamma_t = i, \boldsymbol{\mu}_i, \boldsymbol{\Lambda}_i)$, so I use VB to get $q(\boldsymbol{y}_t)$ and use this to update our estimated posterior $q(\boldsymbol{\mu}_i, \boldsymbol{\Lambda}_i)$.

There are a few parameters to set in this model. Beside π, they are all uninformative. I choose to set $\alpha = 1$, $\boldsymbol{\mu}_0 = 0$, $\nu_0 = 1$, $\kappa_0 = .1$, $\pi = 1e-5$, and $\Phi_0 = .1\mathbf{I}_K$ (the scaled identity matrix).

2.1 Additional Inference for time-evolving mean

Because we are using an exponential kernel, we have the property that for a generic time series $\boldsymbol{x}$ that $x_a|x_b \perp\!\!\!\perp x_c \forall b < a, c < b$. We can use a VB approximation to forward filtering algorithm. Therefore, we can write:

$$p(\mu_{ikt}|\mu_{ik(t-a)}) = \mathcal{N}\left((1-e^{-\beta|a|})\mu_{0k} + e^{-\beta|a|}\mu_{ik(t-1)}, \tau^{-1}(1-e^{-2\beta|a|})\right) \quad (34)$$

If we know that the VB approximation of $q(\mu_{ik(t-a)}) \simeq p(\mu_{ik(t-a)}|\{x_{kn}\}_{n=1}^{t})$, then we get:

$$q(\mu_{ik(t-a)}) = \mathcal{N}(\hat{\mu}_{ik(t-a)}, r) \quad (35)$$

$$p(\mu_{ikt}|\{x_{kn}\}_{n=1}^{t-a}) \simeq \mathcal{N}\big((1-e^{-\beta|a|})\mu_{0k} + e^{-\beta|a|}\hat{\mu}_{ik(t-a)}, \tau^{-1}(1-e^{-2\beta|a|}) + re^{-2\beta|a|}\big) \quad (36)$$

$$(37)$$

If we know the variational distribution for y_{tk} we can get a variational approximation for $p(\mu_{ikt}|, \{x_{kn}\}_{n=1}^{t})$ by using the approximation for $p(\mu_{ikt})$ above. This is trivially extended to the vector case.

3 Results

I ran this on the HC-1 d533101 dataset. It is a 105 second recording that has a tetrode (so 4 extracellular channels) and one intracellular electrode. The extracellular recordings were high-pass filtered at 800Hz. After spike detection, we say that a detected spike corresponds to the intracellular neuron if the spike detection is within .5ms of an intracellular spike (the IC signal is very clean–this are known quite well).

There is some issues with how to present the results and provide informative metrics. There are some error metrics we can use. The first metric is the metric used by Qisong and Bo, where we define the cluster with the greatest number of intracellular spikes as the "intracellular cluster" (IC cluster). Then this error metric can be stated as:

$$e_1 = \frac{\#\text{of IC spikes in the IC cluster} + \#\text{of non-IC spikes in all of clusters}}{\#\text{total detections}} \tag{38}$$

This worked in previous papers because the total number of detections and the number of IC spikes was set, whereas since we're combining detection and sorting, this may be a questionable error metric. (Namely, as (#total detections) $\rightarrow \infty$ we expect that (#of non-IC spikes in all of clusters) $\rightarrow \infty$ so that in the limit $e_1 \rightarrow 1$.)

A second error metric we can use is based only on the "intracellular cluster," which could simply be:

$$e_2 = \frac{\#\text{of IC spikes in the IC cluster}}{\#\text{of total spikes in the IC cluster}} \tag{39}$$

We can also look at spread of the IC spikes and where we are clustering them, so we could define a third qauntity:

$$e_3 = \frac{\#\text{of IC spikes in the IC cluster}}{\#\text{total IC spikes}} \tag{40}$$

Since we are also documenting the detection of the alogorithms, we should also think about how to report the detections. In my view, there are really 2 alternatives. The first is the total number of the intracellular spikes that we consider detected. The second is simply the number of intracellular spikes in the intracellular cluster. Simply stated:

$$n_1 = \#\text{IC spikes detected} \tag{41}$$
$$n_2 = \#\text{IC spikes in the IC cluster} \tag{42}$$

I personally prefer n_2. The reason for this is that as we detect more and more spikes (say π gets small), then we will detect every IC spike, but it is important to both correctly detect the spike and have it be useful information (I.E. in a single cluster) than to be able to say we get everything without it being useful. The results for all of these metrics are shown in Table 1. Figures 1-4 show the clustering in the inferred feature space for the various algorithms.

3.1 Verification of the moving mean property of Neuron firing

As in the Paninski paper, the mean of the neuron appears to move. See Figures 5-7.

4 Things to do

- More datasets.

Algorithm	e_1	e_2	e_3	n_1	n_2
K-means, 3PC, K=2	.5601	.3118	.9827	753	740
K-means, 3PC, K=3	.9433	.8216	.9177	753	691
K-means, 3PC, K=4	.9439	.8236	.9177	753	691
EM-GMM, 3PC, K=2	.7213	.4177	.9774	753	736
EM-GMM, 3PC, K=3	.9335	.7620	.9734	753	733
EM-GMM, 3PC, K=3	.9340	.7641	.9721	753	732
Online-Threshold	.9449	.8257	.9184	753	687
Online-White Noise	.9214	.8141	.7914	809	683
Online-GP Noise	.9426	.7965	.9873	815	775
Online-AR mean and White Noise	.7986	.5113	.9812	839	834
Online-AR mean and GP Noise	.9472	.8267	.9748	819	811

Table 1: Results on HC-1 d533101

- Multi-Channel? I actually did implement the model in Section 1.3 to do multi-channel analysis. It didn't work–as it turns out, the background "noise" is correlated between channels, and I will need to account for this correlation or suffer large numbers of detections; assumedly some of those detections would be false positives. This wouldn't be the hardest thing to do; it is possible to use a Multi-Task GP to do this, but it would take me a couple of days. The speed of the algorithm would also suffer because I would have to invert at PC matrix instead of just a P matrix.
- Sparsely firing neurons? Because the model seems to be good at this based on the results, it may to worthwhile to simulate some data and show that we can capture sparsely-firing neurons.
- Other algorithm comparisons? I have the code for the Kalman-filter mixture model and I will run that, but it's not an online algorithm.

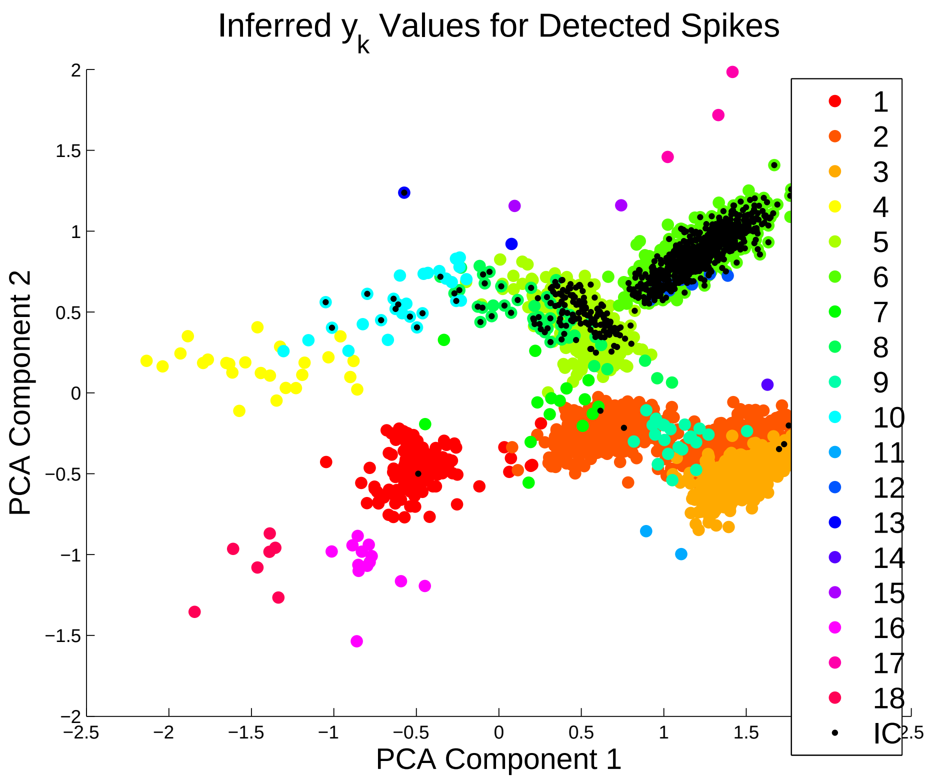

Figure 1: Inferred y_t for detected spikes, online method with white noise model

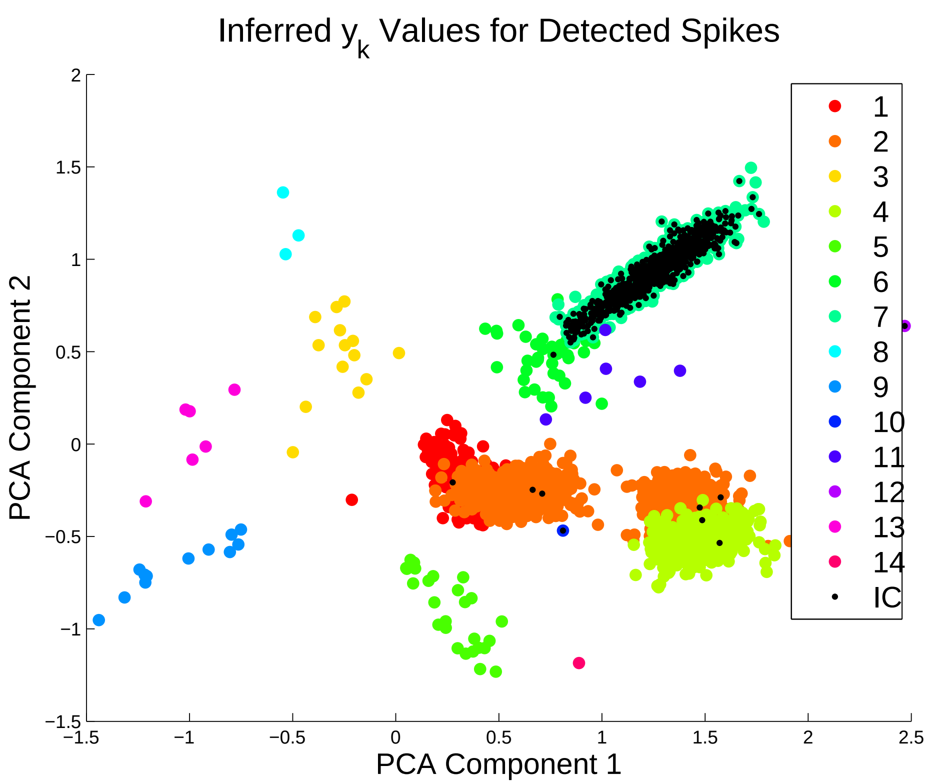

Figure 2: Inferred y_t for detected spikes, online method with GP noise model

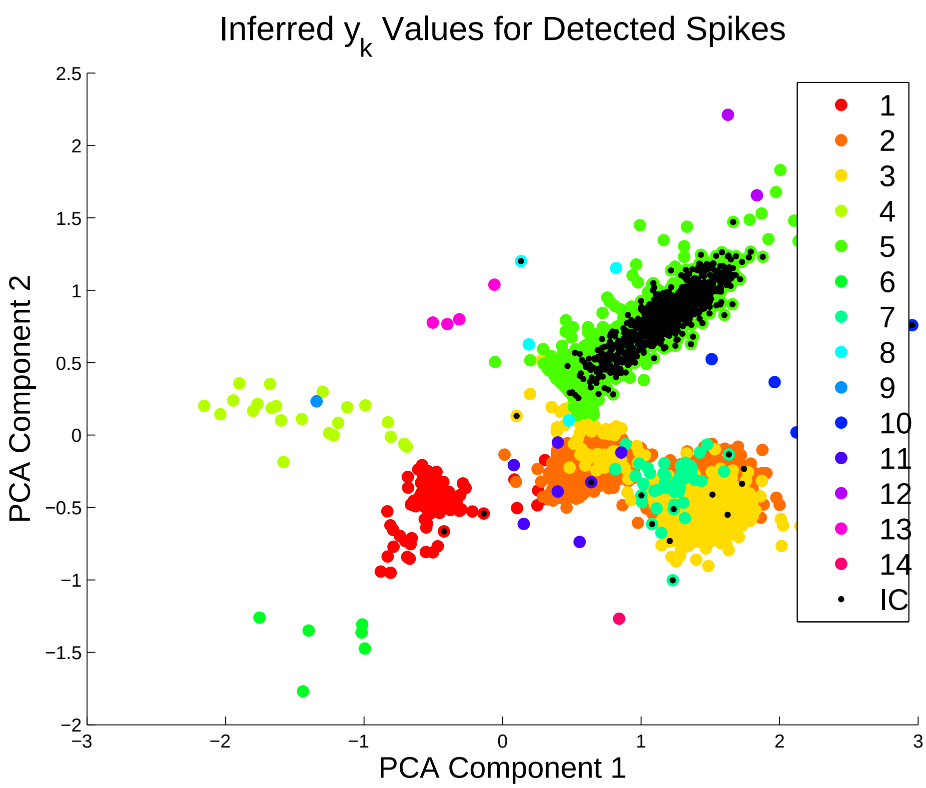

Figure 3: Inferred y_t for detected spikes, online method with GP mean with white noise model

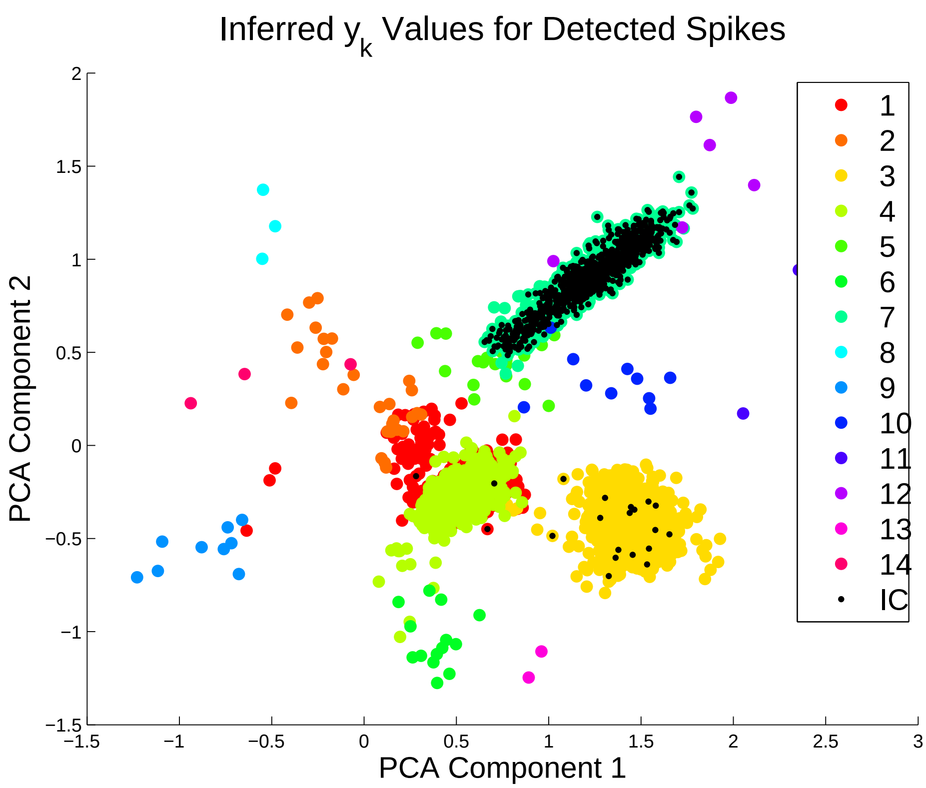

Figure 4: Inferred y_t for detected spikes, online method with GP mean and GP noise model

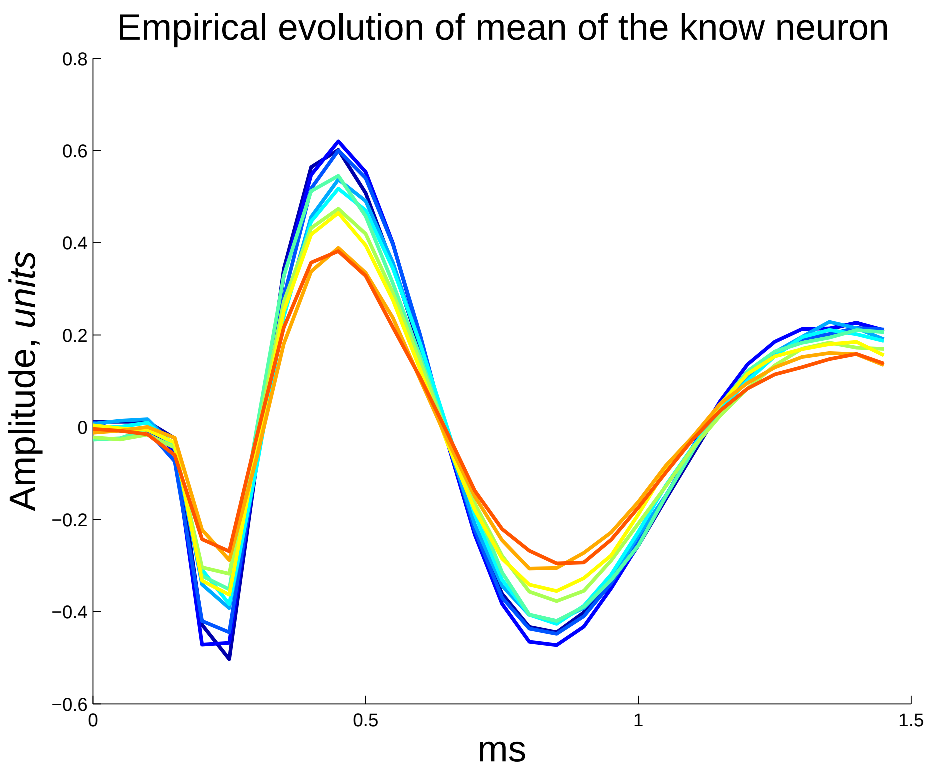

Figure 5: Mean of IC spikes over Time

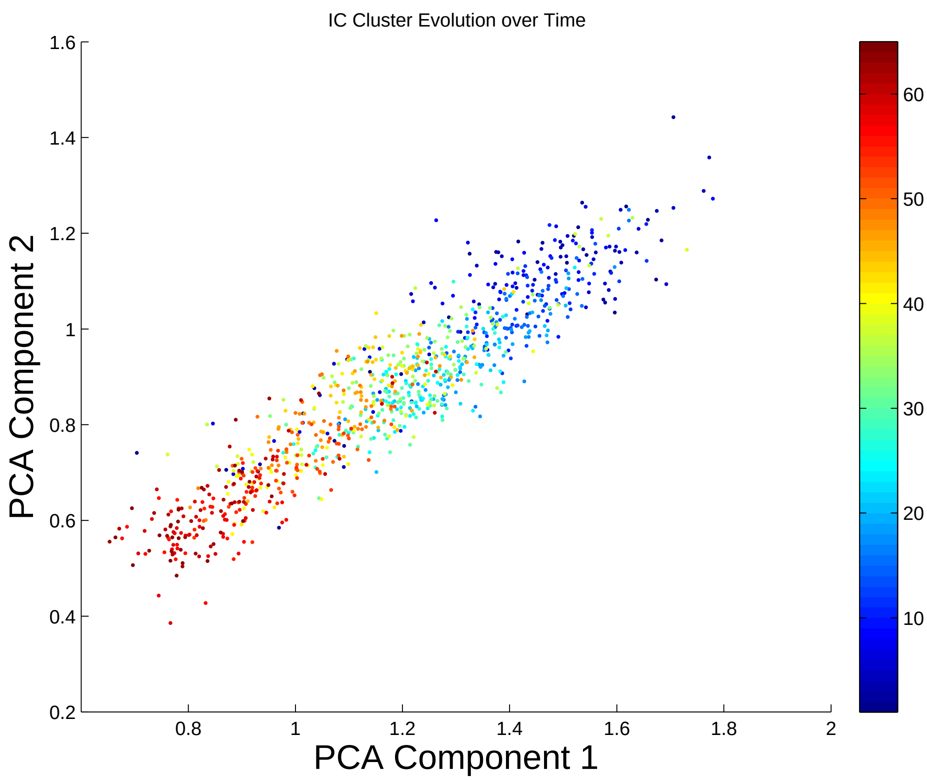

Figure 6: Evolution of the IC cluster in the Online-GP mean and GP noise model

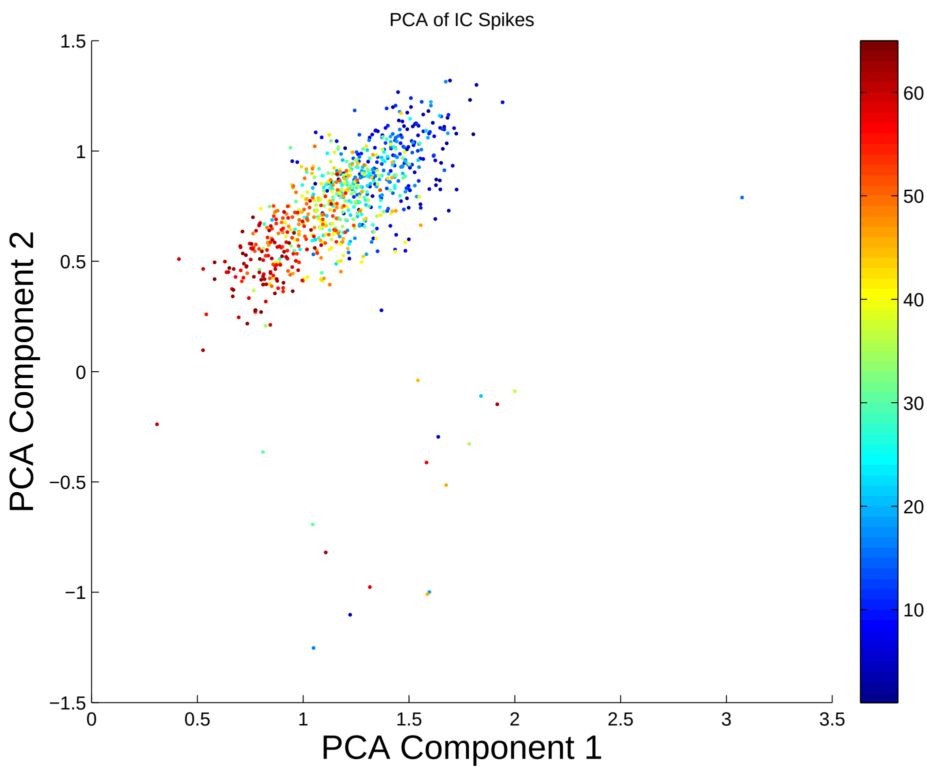

Figure 7: PCA for IC spikes detected with thresholding