Cognitive Neurodynamics (2022) 16:1393–1405
https://doi.org/10.1007/s11571-022-09781-7

RESEARCH ARTICLE

Spike Sorting of Non-Stationary Data in Successive Intervals Based on Dirichlet Process Mixtures

Foozie Foroozmehr[1] · Behzad Nazari[1] · Saeed Sadri[1] · Reyhaneh Rikhtehgaran[2]

Received: 4 August 2020 / Revised: 10 December 2021 / Accepted: 8 January 2022 / Published online: 9 February 2022

Abstract

This paper proposes a new automatic method for spike sorting and tracking non-stationary data based on the Dirichlet Process Mixture (DPM). Data is divided into non-overlapping intervals and mixtures are applied to individual frames rather than to the whole data. In this paper, we have used the information of the previous frame to estimate the cluster parameters of the current interval. Specifically, the means of the clusters in the previous frame are used for estimating the cluster means of the current one, and other parameters are estimated via noninformative priors. The proposed method is capable to track variations in size, shape, or location of clusters as well as detecting the appearance and disappearance of them. We present results in two-dimensional space of first and second principal components (PC1-PC2), but any other feature extraction method leading to the ability of modeling spikes with Normal or t-Student distributions can also be applied. Application of this approach to simulated data and the recordings from anesthetized rat hippocampus confirms its superior performance in comparison to a standard DPM that uses no information from previous frames.

Keywords Bayesian clustering · Dirichlet process mixture · Non-stationary data · Spike sorting

Introduction

There have been many researches studying electrical signals of neurons to understand brain functionality. Analyzing a single neuron activity leads to a limited view, and hence, the activities of neuron populations are investigated instead (Brown et al. 2004). Using Multi-Electrode Arrays (MEAs) for recording tens or even thousands of channels simultaneously is one solution (Navratilova et al. 2015), and the signals recorded by MEAs must be divided into single neuron activities. In other words, the firing patterns of each neuron should be extracted. "Spike sorting" is a procedure that extracts each neuron activity from background noise and other nearby neurons, working in a population to form a particular action. Thus, it is an essential procedure to characterize the firing properties of individual neurons (Rodrigo Quian Quiroga 2012). Accordingly, this process is needed to understand the brain's electrical circuitry and is also used as the Brian Machine Interface (BMI) inputs (Gibson et al. 2010) (See e.g. Lewicki 1998; Lefebvre et al. 2016; Zamani et al. 2020, to review some spike sorting methods).

Basically, spike sorting consists of the following steps (Huang et al. 2021): first, the recorded signals are bandpass filtered (Rey et al. 2015). Then, spike occurrence times are determined by a process called "detection" (See e.g. Shahid et al. 2010; Yuan et al. 2012; Quiroga et al. 2004, for some detection approaches). In the next step named "feature extraction", detected spikes are projected into feature space; and if necessary, dimensionality reduction is done to make further processes simpler and faster (some feature extraction methods: Wu and Swindlehurst 2018; Paraskevopoulou et al. 2013; Zamani and Demosthenous 2014; Kamboh and Mason 2012; Soleymankhani and Shalchyan 2021). In the "Clustering" step, similar spikes in terms of their shapes are grouped into clusters so that each cluster corresponds to an individual neuron.

However, spike sorting or in particular, the clustering step, may be performed in a different way called template-

✉ Behzad Nazari
nazari@cc.iut.ac.ir

[1] Department of Electrical and Computer Engineering, Isfahan University of Technology, 84156-83111 Isfahan, Iran

[2] Department of Mathematical Sciences, Isfahan University of Technology, 84156-83111 Isfahan, Iran

matching-based method. In some methods, template matching is used to save time in the clustering step, e.g. in ‘Wave Clus’ (Quiroga et al. 2004), in which the obtained templates from the subset of data are used for clustering the remainder spikes. Some other methods apply a preliminary clustering step to get initial templates and use them to reconstruct raw data (Lee et al. 2020; Garcia and Pouzat 2019; Yger et al. 2018). This procedure makes it possible to find the overlapped spikes that were not detected properly in the detection step. A similar approach can be performed after the clustering step of the basic methods to improve the performance. Another interesting method is proposed in (Pachitariu et al. 2016), in which all the parameters i.e., spike times, spike amplitudes, templates, and cluster assignments are optimized via minimization of a particular cost function. Therefore, this method does not have the feature extraction step by working with raw data. In addition, detection and clustering steps are performed simultaneously.

One of the spike sorting challenges is the variations of the spike waveforms through time when the duration of recording becomes long. Different factors like electrode drifts, variable background noise, and variations of spike shape characteristics lead to variations in recorded spike waveforms over time (Bar-Hillel et al. 2006). Moreover, neuron death or recording the activities of a new neuron due to electrode drifts is possible (Gasthaus and Wood 2008). Finally, these events may appear in feature space in different ways: (i) drifts of clusters and change of their shapes, (ii) split of one cluster or merging two different clusters into one, (iii) creation of a new cluster or disappearance of one.

In order to track these variations in the clustering step, different solutions have been proposed. Some previous Bayesian researches assume data to be stationary for a certain interval, Normal mixture model is considered for the period (Bar-Hillel et al. 2006; Wolf and Burdick 2009; Calabrese and Paninski 2011; Gasthaus et al. 2009) and then, a clustering method is applied to track some of the so-called variations. In (Bar-Hillel et al. 2006) and (Wolf and Burdick 2009), in order to recognize the true number of clusters, a few candidate models with different neurons are assumed. After dividing data into frames with specified duration by Bar-Hillel et al., some candidate mixture models are computed for each interval. Then, based on the considered Bayesian network model for data generation, transition probabilities between candidate mixtures are computed, and eventually, the best models for the whole data are obtained as the MAP solution of the final probabilistic model. Unlike this approach which cannot be implemented in real-time, Wolf et al. use just the previous frame information for the current interval. Indeed, priors of cluster means in frame $(t + 1)$ are considered to have the distribution of Normal mixture, centered at cluster means of frame (t) plus uniform distribution. With this definition for mean priors, it would be possible to track changes in cluster numbers. However, this method has a relatively large complexity in some steps, such as the adjustment of cluster numbers in expectation–maximization (EM) initial states with their number in the previous frame.

A more simple method is proposed in (Calabrese and Paninski 2011), which assumes the mean of cluster ‘g’ in the time $(t + 1)$ equals the mean of that cluster in the previous time step plus Gaussian noise. Therefore, cluster center changes are modeled via Normal distribution with the conservative assumption that the number of clusters is known initially and does not change through time.

Another method (Gasthaus et al. 2009) uses Dirichlet process mixture instead of applying candidate models with different cluster numbers, and therefore, it estimates the number of mixture components as well as their parameters. In this model, the parameters are shared during time steps and results show a high False Negative rate when data is stationary. In addition, in this approach, the amount of the Dirichlet process concentration parameter is selected under supervision. Therefore, there is no guarantee that the algorithm works well when is applied to other real data.

Table 1 reviews the comparison among the above-mentioned methods. This table actually has two parts; the first part compares the features and the second part compares the capabilities of different methods.

On the first part Different distributions can be assumed for observed data but the most common one is the Normal distribution (Bar-Hillel et al. 2006; Wolf and Burdick 2009; Calabrese and Paninski 2011; Gasthaus et al. 2009). In (Wolf and Burdick 2009) and (Calabrese and Paninski 2011) cluster drifts are modeled by assuming Normal distribution for the prior of current cluster means. These Normal Distributions are centered at the cluster means of the previous time. This assumption is not used in (Bar-Hillel et al. 2006) and (Gasthaus et al. 2009). Data is divided into shorter intervals in (Bar-Hillel et al. 2006) and (Wolf and Burdick 2009) so that the interval stationarity assumption becomes reasonable. But the other two methods (Calabrese and Paninski 2011; Gasthaus et al. 2009), process the data after a new spike is detected. In (Calabrese and Paninski 2011), the number of clusters is estimated first and it is assumed to be fixed through time. Methods in (Bar-Hillel et al. 2006) and (Wolf and Burdick 2009), select the cluster number of each frame from some candidate models with different numbers of clusters which adds more complexity to the model. On the other hand, the Dirichlet process mixture is used to estimate the number of clusters in (Gasthaus et al. 2009).

On the second part The possibility of real-time implementation is an important capability of an algorithm.

Table 1 Comparison among some of the most important bayesian methods of spike sorting

Properties	Bar-Hillel et al. (2006)	Wolf and Burdick (2009)	Calabrese and Paninski (2011)	Gasthaus et al. (2009)
Normal distribution for data	×	×	×	×
t-Student distribution for data				
Normal cluster mean drifts		×	×	
Dividing data into frames	×	×		
Selecting number of clusters among some candidate models	×	×		
Selecting number of clusters using Dirichlet process mixture				×
Possibility of real-time implementation		×	×	×
Possibility to track changes in cluster numbers	×	×		×
Possibility to track cluster drifts	×	×	×	×
Possibility to track changes in cluster shapes	×	×	×	×

Except (Bar-Hillel et al. 2006) which finds the best mixture model of all frames after processing the whole data, other approaches have the potential of being implemented in real-time or with a delay equal to frame duration. All methods track variations in cluster numbers except (Calabrese and Paninski 2011) that assumes a fixed cluster number through time. All approaches have the possibility of tracking cluster drifts and shape variations.

In this paper, we propose a new method based on the Dirichlet process mixture that utilizes the previous frame information for the current one. Leveraging Dirichlet process mixture properties, the proposed method estimates the number of clusters as well as their parameters, and due to the use of the clustering results of the previous frame, the proposed method performs better than the approaches that cluster frames independently. Therefore, the number of clusters is not required to be known initially and is simultaneously estimated with cluster parameters as well as the concentration parameter of the Dirichlet process, without the need for several candidate models, and try and error procedures, which are considered in the so-called previous approaches. In addition, if there is any change in the number, size, shape, or location of clusters, the proposed method can track them over time.

The remainder of this paper is organized as follows: Method section presents the design procedure of the proposed method. Results of applying proposed method to the simultaneous intra-extra cellular recording as well as simulated data are given in the Results section. In addition, some diagrams showing the performance of our approach are presented. In the last section, concluding remarks are provided.

Method

In this article, our focus is on the clustering step of spike sorting. Therefore, it is assumed that spikes are first detected in a particular way, and then are projected into feature space. We used the first and second principal components (PC1, PC2) as features, but any other feature extraction method can be used instead. Consequently, in the remainder of this paper, we call the points in the feature space as “spikes”. These spikes are the input for the clustering step. Our emphasis is on Bayesian clustering as well as tracking the non-stationarity nature of spikes through time. If we show spike points with $\{\boldsymbol{v}_i\}_{i=1}^{L}$, with L be the number of spikes, in the structure of model-based clustering we use the following mixture distribution:

$$p(\boldsymbol{v}_i) = \sum_{k=1}^{K} \pi_k f(\boldsymbol{v}_i|\boldsymbol{\theta}_k) \tag{1}$$

in which K denotes the number of clusters or equivalently, the number of individual neurons, π_k’s ($k = \{1, \ldots, K\}$) denote mixture probabilities with $\sum_{k=1}^{K} \pi_k = 1, \pi_k \geq 0$ for all k, and f is a probability density function with $\boldsymbol{\Theta} = \{\boldsymbol{\theta}_k\}_{k=1}^{K}$ as its parameters. Most of the time f is assumed to be a Normal distribution and hence $\boldsymbol{\theta}_k$ is $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$, mean vector and covariance matrix of the k$^{\text{th}}$ component of the mixture, respectively. It must be mentioned that using other distributions such as the t-Student distribution is also possible for f, as in Shoham et al. (2003). In the case of using the t-Student distribution, we have: $\boldsymbol{\theta}_k = \{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \vartheta_k\}$, where $\boldsymbol{\mu}_k$, $\boldsymbol{\Sigma}_k$ and ϑ_k are respectively the location, scale, and the shape parameter of the distribution. We used both Normal and t-Student distributions for f, and for t-Student distribution, the hierarchical representation was applied, i.e. the relation between t-Student, Normal and Gamma distributions as follows:

$$t(\boldsymbol{v}_i|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \vartheta_k) = \int_0^\infty N(\boldsymbol{v}_i|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k/w)G(w|\vartheta_k/2, \vartheta_k/2)dw \tag{2}$$

where t, N and G are t-Student, Normal and Gamma distributions, respectively.

In Bayesian point of view, model parameters ($\boldsymbol{\Theta} = \{\boldsymbol{\theta}_k\}_{k=1}^K$) are random variables having probability density functions with parameters $\boldsymbol{\theta}_{H_k}$, called "hyper-parameters":

$$\boldsymbol{\theta}_k \sim g(\boldsymbol{\theta}_{H_k}) \tag{3}$$

Here, K should be determined a priori, that is an issue for complex or real data.

One approach to estimate the cluster parameters is using the Dirichlet process mixtures (DPM). DPM uses the Dirichlet process (DP) as priors for mixture parameters, thus the result is estimating parameters as well as the number of mixture components. In the spike sorting problem, where knowing the true number of clusters is very important, DPM has been used in (Wood and Black 2008). But variations of clusters through time is another challenge that should be considered.

The main point of using the Dirichlet process mixture is finding the true number of K which best fits the data. Therefore, if $\{\boldsymbol{\theta}_i^*\}_{i=1}^L$ indicate the parameters corresponding to the data points $\{\boldsymbol{v}_i\}_{i=1}^L$, we have:

$$\begin{aligned} &\boldsymbol{v}_i|\boldsymbol{\theta}_i^* \sim f(\boldsymbol{v}_i|\boldsymbol{\theta}_i^*) \\ &\boldsymbol{\theta}_i^*|\mathbb{G} \sim \mathbb{G} \\ &\mathbb{G}|\alpha, \mathbb{G}_0 \sim \mathrm{DP}(\alpha, \mathbb{G}_0) \end{aligned} \tag{4}$$

where $\mathbb{G}$ is a discrete distribution (thus multiple $\boldsymbol{\theta}_i^*$'s can take on the same value simultaneously), $\mathbb{G}_0$ is the base distribution and α is the concentration parameter of the DP. Actually, $\mathbb{G}_0$ is the mean of the Dirichlet process and α has an inverse relationship with the variance of the DP. This means that the concentration of mass around $\mathbb{G}_0$ is controlled by α (Teh 2011).

Based on the clustering property of the DP, finite samples of $\mathbb{G}$, i.e., $\{\boldsymbol{\theta}_i^*\}_{i=1}^L$'s, will share repeated values with positive probability. We define these unique values of $\boldsymbol{\theta}_1^*, \ldots, \boldsymbol{\theta}_L^*$ as $\varphi_1, \ldots, \varphi_K$.

Using the "Stick-Breaking" (SB) representation of the Dirichlet process (Sethuraman 1994), sampling from $\mathbb{G}$ distribution becomes feasible. Therefore, if $\{\varphi_i\}_{i=1}^K$ are the shared values of $\boldsymbol{\theta}_i^*$'s, then in the SB representation of the DP, we have:

$$\mathbb{G}(\boldsymbol{\theta}_i^*) = \sum_{j=1}^\infty \pi_j \delta_{\varphi_j}(\boldsymbol{\theta}_i^*) \tag{5}$$

$$, \begin{cases} \varphi_j \sim \mathbb{G}_0 \\ \pi_j = V_j \prod_{r=1}^{j-1}(1 - V_r), V_j \sim Beta(1, \alpha) \end{cases} \tag{6}$$

where δ is the Dirac delta function and $Beta$ is the Beta distribution with the probability density function of (7), having shape parameters of $\beta, \alpha > 0$:

$$Beta(x; \beta, \alpha) = \frac{x^{\beta-1}(1-x)^{\alpha-1}}{\int_0^1 u^{\beta-1}(1-u)^{\alpha-1}du}, \tag{9}$$

For the base distribution, $\mathbb{G}_0$ we have:

- When using Normal distribution for data with cluster parameters $\boldsymbol{\mu}_j$'s and $\boldsymbol{\Sigma}_j$'s (cluster means and covariance matrices, respectively), we assume that the parameters are independent and therefore:

  $$\mathbb{G}_0 = \mathbb{G}_0^{\mu_j} \times \mathbb{G}_0^{\Sigma_j} \tag{8}$$

  where $\mathbb{G}_0^{\mu_j}$ and $\mathbb{G}_0^{\Sigma_j}$ are the prior distributions of $\boldsymbol{\mu}_j$'s and $\boldsymbol{\Sigma}_j$'s respectively. Here we set $\mathbb{G}_0^{\mu_j}$ as Normal and $\mathbb{G}_0^{\Sigma_j}$ as inverse-Wishart distributions.
- When using t-Student distribution, with parameters $\boldsymbol{\mu}_j$'s, $\boldsymbol{\Sigma}_j$'s, and ϑ_j's (location, shape, and scale parameters, respectively), with the assumption of parameter independence, we have:

  $$\mathbb{G}_0 = \mathbb{G}_0^{\mu_j} \times \mathbb{G}_0^{\Sigma_j} \times \mathbb{G}_0^{\vartheta_j} \tag{9}$$

Here we set Normal, inverse-Wishart and Gamma distributions for $\mathbb{G}_0^{\mu_j}$, $\mathbb{G}_0^{\Sigma_j}$, and $\mathbb{G}_0^{\vartheta_j}$, respectively.

In addition, the Gamma distribution (Gamma(1,1) as in (Wood et al. 2006)) is considered as prior distribution of α.

Tracking the cluster locations is the most important task in neuron tracking, thus our emphasis is on $\boldsymbol{\mu}_j$ and its prior ($\mathbb{G}_0^{\mu_j}$). We use DPM for each individual time interval of data and the main point is using the result of the previous frame for the current one. $\mathbb{G}_0^{\mu_j}$ is assumed to be Normal and hence its hyper-parameters are the mean vector and the covariance matrix. We propose to use $\boldsymbol{\mu}_j$'s of the previous frame as the mean vector hyper-parameter of the current frame mean base distribution. Therefore, tracking spike waveform variations through time becomes possible. Consequently, our method has one form for $t = 1$ and another form for $t > 1$:

For $t > 1$, three different cases would happen:

A. If the number of clusters does not change in frame (t) with respect to frame (t-1):

   There are three possible cases: 1) One or more clusters of frame (t) change their size, shape or location in the feature space, in comparison to frame (t-1), 2) Clusters do not change in comparison to the previous frame, and 3) a new cluster (neuron) is found while another cluster (neuron) is lost.

In the first two cases, the cluster mean of frame (t) can be assumed to be the mean of the previous frame plus a Gaussian noise; therefore, $\boldsymbol{\mu}_j^t$'s (cluster means of frame (t)) can be generated with a normal distribution centered at $\boldsymbol{\mu}_j^{t-1}$'s with compressed support in comparison to data dispersion according to SB representation of the DP (Eq. (6)).

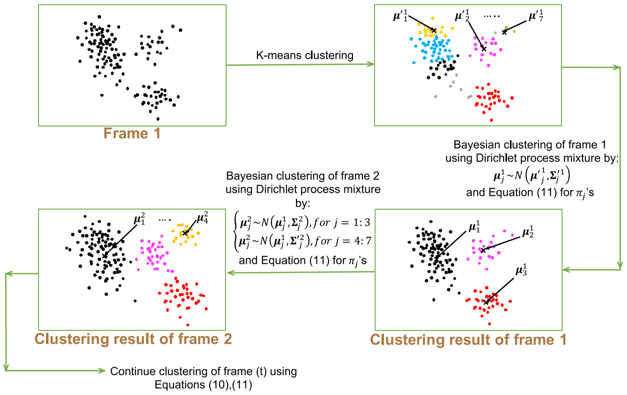
K-means clustering
Frame 1
$\boldsymbol{\mu}'^1_1$
$\boldsymbol{\mu}'^1_2$
.....
$\boldsymbol{\mu}'^1_7$
Bayesian clustering of frame 1 using Dirichlet process mixture by:
$\boldsymbol{\mu}_j^1 \sim N\left(\boldsymbol{\mu}'^1_j, \boldsymbol{\Sigma}'^1_j\right)$
and Equation (11) for π_j's
Bayesian clustering of frame 2 using Dirichlet process mixture by:
$\begin{cases} \boldsymbol{\mu}_j^2 \sim N(\boldsymbol{\mu}_j^1, \boldsymbol{\Sigma}_j^2), for\ j = 1:3 \\ \boldsymbol{\mu}_j^2 \sim N(\boldsymbol{\mu}_j^1, \boldsymbol{\Sigma}'^2_j), for\ j = 4:7 \end{cases}$
and Equation (11) for π_j's
$\boldsymbol{\mu}_1^2$
....
$\boldsymbol{\mu}_4^2$
$\boldsymbol{\mu}_1^1$
$\boldsymbol{\mu}_2^1$
$\boldsymbol{\mu}_3^1$
Clustering result of frame 2
Clustering result of frame 1
Continue clustering of frame (t) using Equations (10),(11)

Fig. 1 A schematic of clustering procedure for successive frames. Starting from K-means clustering of the first frame, the result is used for clustering of frame 1 via the proposed method. Utilizing the obtained result, clustering of frame 2 is accomplished, and this procedure continues till the last frame

In fact, the cluster mean drifts are modeled via Gaussian noise and the cluster shape and size variations are also trackable with the estimated covariance matrix of data. Therefore we can generate $\boldsymbol{\mu}_j^t$'s with $\mathbb{G}_0^{\mu_j}$ (Eq. (8)).

The third case is involved in the following conditions, simultaneously.

B. If new clusters are generated:

The means of new clusters can be generated using wide supported Normal distribution providing non-informative prior, in order to let new clusters to be located anywhere in the feature space. Therefore, $\boldsymbol{\mu}_j^t$'s of new clusters can be generated using $\mathbb{G}_0^{\mu_j}$ with zero mean vector and a covariance matrix leading to a wide supported distribution (compared with data dispersion) according to (6) and (8).

C. If some neurons have lost:

When generating π_j's (Eq. (6)), those corresponding to the lost neurons would be very small that they should be neglected; that is the sign of disappearance of the relevant clusters.

We define the maximum possible number of clusters in the t^th^ frame to be K^t. In each frame, the true number of clusters is estimated using the Dirichlet process mixture. We recall the true number of clusters as M^t ($M^t \leq K^t$). Now for estimating the cluster means of the current frame, for

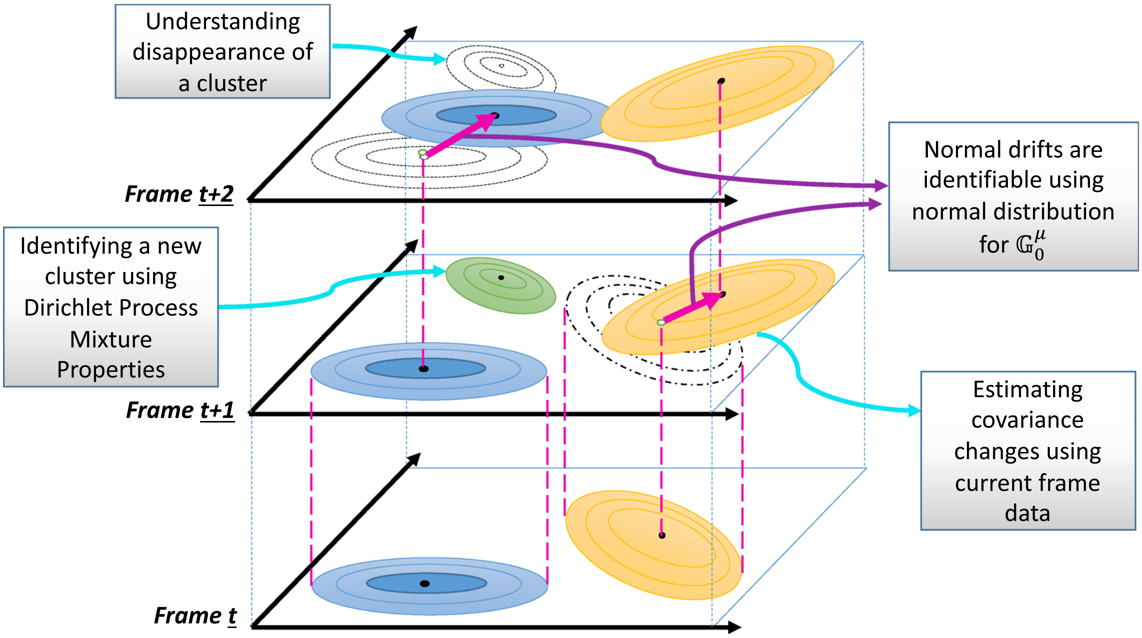
Understanding disappearance of a cluster
Normal drifts are identifiable using normal distribution for $\mathbb{G}_0^{\mu}$
Identifying a new cluster using Dirichlet Process Mixture Properties
Estimating covariance changes using current frame data
Frame t+2
Frame t+1
Frame t

Fig. 2 Properties of the proposed method. Any change in the number of clusters, normal drifts as well as variations in the covariance matrix (or in other words, cluster shapes) are identified via the proposed method

neuron numbers from 1:M^{t-1} (M^{t-1} is the number of previous frame neurons) we use Normal distributions centered at cluster means of frame (t-1), and in order to find newly generated neurons, $K^t - M^{t-1}$ wide supported Normal distributions are used. Finally, our model for frame $t > 1$will be:

$$\begin{cases} \boldsymbol{\mu}_j^t \sim N\left(\boldsymbol{\mu}_j^{t-1}, \boldsymbol{\Sigma}_j^t\right) \text{ for } j = 1 : M^{t-1} \\ \boldsymbol{\mu}_j^t \sim N\left(0, \boldsymbol{\Sigma}_j'^t\right) \text{ for } j = M^{t-1} + 1 : K^t \end{cases} \tag{10}$$

$$, \begin{cases} \pi_j = V_j \prod_{r=1}^{j-1} (1 - V_r) \\ V_j \sim Beta(1, \alpha) \end{cases} \text{ for } \quad j = 1 : K^t \tag{11}$$

According to the dispersion of data, $\boldsymbol{\Sigma}_j^t$ should be chosen in such a way that in (10), the Normal distribution $\mathbb{G}_0^{\mu_j^t} = N\left(\boldsymbol{\mu}_j^{t-1}, \boldsymbol{\Sigma}_j^t\right)$, has compressed support. But, $\boldsymbol{\Sigma}_j'^t$ should lead to a wide supported distribution in $\mathbb{G}_0^{\mu_j^t} = N\left(0, \boldsymbol{\Sigma}_j'^t\right)$ providing non-informative prior. Moreover, the inverse-Wishart distribution is used for priors of covariance matrices.

Finally, for frame t = 1, there is no previous frame, so we use the results of K^1-means clustering with a large enough K^1 (we used 7), as priors for $\left\{\boldsymbol{\mu}_j^1\right\}_{j=1}^{K^1}$. Figure 1 shows a schematic of the clustering procedure for successive frames.

Using this model, the information of the previous frame is used only for cluster means of the current frame; and for covariance matrices, the model is free to find the best amounts according to current frame data by using noninformative priors. The outcome would be the possibility to track changes in the number of clusters as well as variations in size, shape, or location of them. Figure 2 summarizes our method properties.

Table 2 Results of clustering two-dimensional generated data using DP-N

Frame number	True number of clusters	Estimated number of clusters using DP-N	Error
1	3	3	0
2	3	3	0
3	3	3	0
4	4	4	0
5	4	4	0.75%
6	3	3	1%

Results

In the first subsection, we show the properties of our proposed method by applying it to our generated two-dimensional data. Then in the next two subsections, we evaluate our suggested approach by applying it to two datasets, one real and one simulated data.

Investigating properties of the proposed method

In order to demonstrate the properties of the proposed method, we have generated two-dimensional data as depicted in Fig. 3. Arrows in Fig. 3b show the direction of cluster drifts, and it is clear that there has been a cluster generation in frame 4, and neuron loss is happened in frame 6 compared to frame 5. Clusters have different covariance matrices in a frame, and the covariance matrix of cluster 3 changes from frame1 to frame 4. Moreover, cluster 2 moves and gradually overlaps with cluster 1. We have applied the proposed method with Normal data distribution (we call it DP-N) to the generated data. Results showed only 0.3% error in the whole data, together with true cluster number estimation, and well-tracking drift and covariance variations. Table 2 summarizes the results of clustering this data using DP-N.

Clustering of partially labeled data

Models were applied to a subset of simultaneously intra-extra cellular recording. These data are publicly available at http://crcns.org/data-sets/hc/hc-1 and are described in (Henze et al. 2009). The data consist of recordings from a tetrode, containing extracellular (EC) data and an intracellular (IC) electrode simultaneously. This dataset is called "partially labeled" because the only neuron having label in EC channels is the one that is also recorded by the IC electrode. Data from tetrode include action potentials of labeled or identified neuron, as well as other neurons surrounding EC electrodes.

The detection step is done according to (Calabrese and Paninski 2011). That is, EC spikes were detected as the local maxima near which the signal exceeds 6 median absolute deviations in magnitude and, IC spikes as the local maxima near which the first derivative of the signal exceeds a certain threshold. Action potentials of the identified neuron in the EC channels were determined as the spikes that occur within 0.1 ms of the IC channel spikes. Each spike is extracted as a vector of 40 samples (19 samples before and 20 samples after the peak). We used PC1 and PC2 as features, therefore each spike is a point in the two-dimensional space of PC1-PC2. Figure 4 shows the

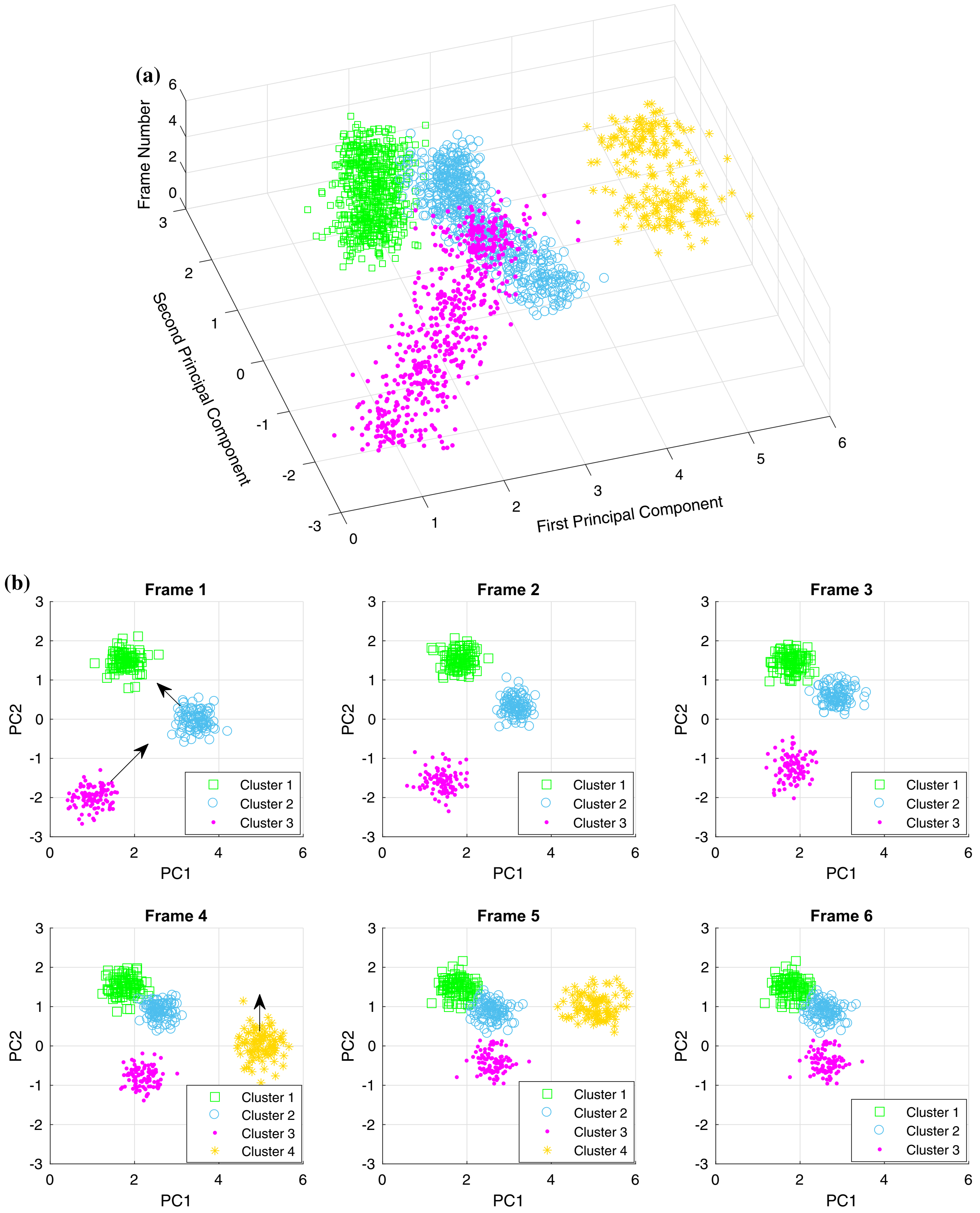

Fig. 3 Two-dimensional generated data for investigating the proposed method properties. **a** cluster variations through time, **b** clusters in individual frames, with arrows showing the direction of cluster drifts

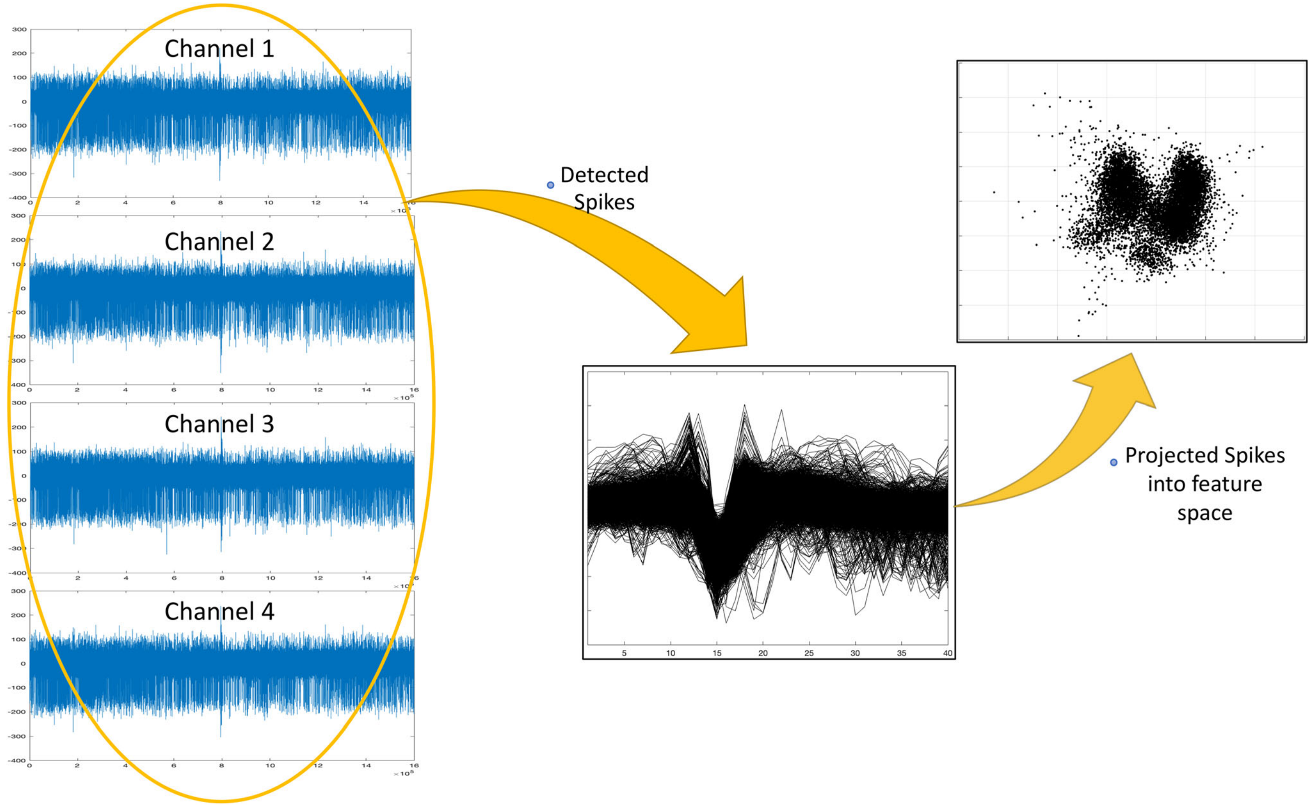

Fig. 4 Filtered signal of 4 EC channel, detected spikes and their projection into the feature space of PC1-PC2

filtered signals of 4 EC channels, the detected spikes, and their projection into feature space.

The clustering step in this paper is done by applying our method to the 16 sequential intervals of data, i.e. applying (10) for mean priors with $K^t = 7$, (11) for mixture probabilities, inverse-Wishart distribution for covariance matrix priors, and 2-dimensional Normal and t-Student distributions for data. We used openBUGS software (Lunn et al. 2009) for defining the data model.

Like (Bar-Hillel et al. 2006) and (Wolf and Burdick 2009), it is assumed that in short intervals, data is stationary, and so Dirichlet process mixtures with their special priors are applied to short intervals instead of applying them to the whole data. We used frames with a duration of 10 s, just the same as Wolf.

Note that, the output of the clustering step of each frame could be used to find templates and be utilized for finding the overlapped spikes which were not detected in the detection step properly, similar to (Lee et al. 2020; Garcia and Pouzat 2019; Yger et al. 2018). Then, another clustering phase could be applied by matching the newly detected spikes with the obtained templates.

For comparison, standard DPM (http://www.robots.ox.ac.uk/∼fwood/code/index.html) is applied to time frames, which is described in (Wood and Black 2008). The difference between our method and Woods is that we have used special prior from the previous frame.

Figure 5 shows the identified neuron spikes and the results of clustering using our method (rows 2 & 3) as well as standard DPM results in row 4 for 12 out of 16 frames. The exact value of False Negative (FN), False Positive (FP), and total error rates are summarized in Fig. 6a, b, and c. The results show that in the proposed DP-N method, the FN rate in 81.25% of frames, the FP rate in 56.25% of frames, and the total error rate in 62.5% of frames, is better than or equal to DPM. But when using t-Student distribution for data (we call it DP-T), in FP and FN rates, this method has similar results with DPM but in total error rates, DPM has a better outcome. Therefore, it can be concluded that Normal distribution for data, when using Dirichlet process as parameter priors, leads to lower error rates.

Note that in frames like 8 and 10, the FN rate of DP-N is much smaller than DPM, but the FP rate is the opposite. This means that the proposed method classifies the

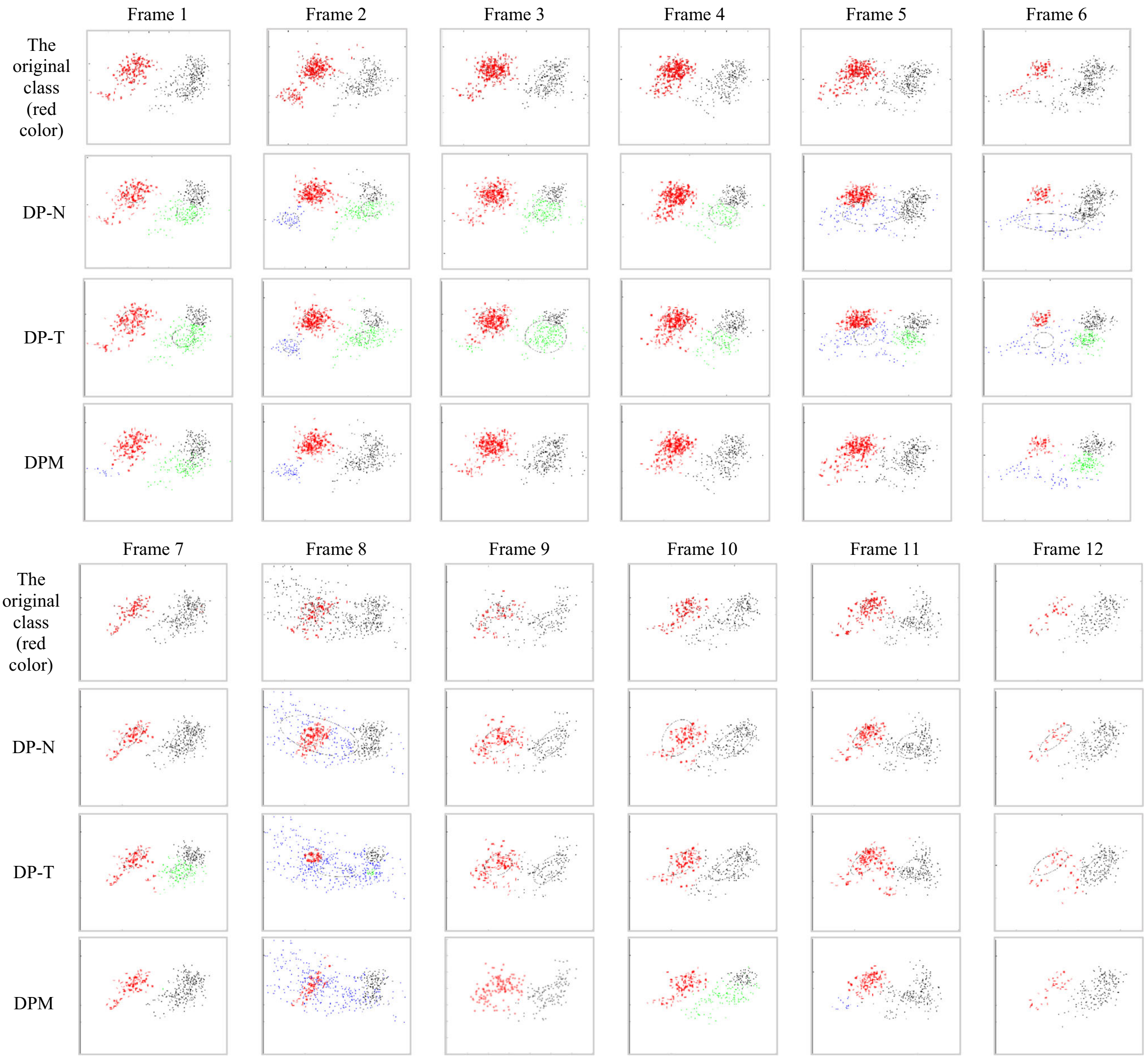

Fig. 5 The identified neuron spikes and the results of clustering using our method (rows 2 & 3) as well as standard DPM results in row 4 for 12 out of 16 frames. The horizontal and vertical axes are the first and second principal components, respectively

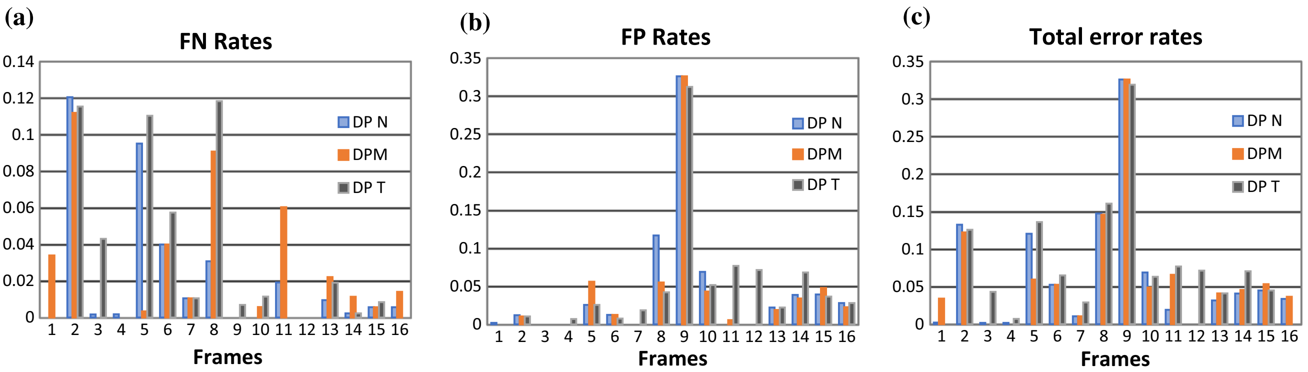

Fig. 6 Different error types. **a** FN, **b** FP and **c** Total error rates of applying DP-N, DPM & DP-T to 16 successive intervals

Table 3 Real data error rates among different methods

Approaches	Mean of FN rate	Mean of FP rate	Mean of total error rate
DP-N	0.0214 ± 0.036	0.0433 ± 0.0818	0.0324 ± 0.0631
DPM	0.0256 ± 0.0344	0.0395 ± 0.0792	0.0326 ± 0.0605
DP-T	0.0322 ± 0.044	0.0487 ± 0.0747	0.0404 ± 0.0609

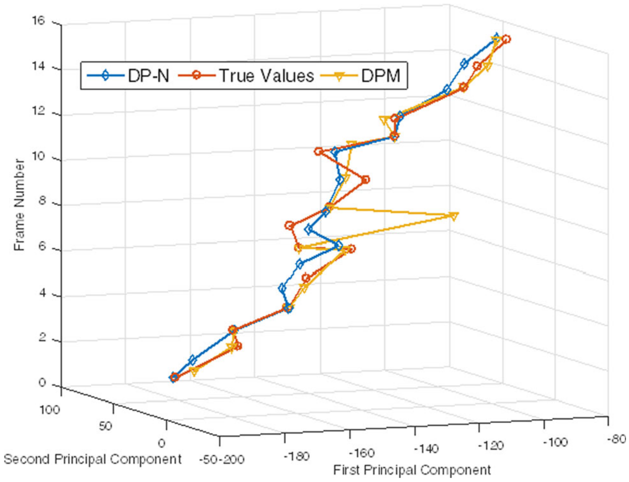

Fig. 7 First and second principal components of cluster means in 16 successive intervals; true values, results of DP_N and DPM methods

Table 4 Comparing cluster mean tracking quality between DP-N and DPM for the real data

Approaches	Mean square error in first component of cluster means	Mean square error in second component of cluster means
DP-N	42.0041	51.9718
DPM	64.5399	52.2716

identified neuron cluster as large as possible, and so the number of other neuron spikes belonging to this cluster increases. But DPM uses clusters with fewer data points for the labeled neuron.

Another point is about the high total error rate in frame 5. In statistics, it is quite justifiable to divide one cluster into two with different dispersions. This is what happened with DP-N in this frame. Although leading to a high total error rate, in statistical view, it is reasonable.

To get a better view of error rates, the mean value and standard deviation of FN, FP, and total error rates are reported in Table 3. According to the table, the weaker performance of DP-T in comparison to other methods is obvious. In addition, DP-N has better performance than DPM in terms of FN and total error rate. The minor difference between these two methods is due to sudden big errors that occur in some frames for both methods.

Now, in order to compare DP-N and DPM (two methods having better results than DP-T), the quality of tracking the cluster means through time is also investigated. Figure 7 shows the first two principal components of cluster means for DP-N, DPM, and their true values in 16 successive intervals. The difference between the points of DP-N and DPM from their true values is calculated and finally, the mean square errors are computed and reported in Table 4.

Clustering of simulated data

Since the real dataset used in the previous subsection was partially labeled, here we evaluate our method via fully labeled simulated data reported in (Buccino and Einevoll 2021). We applied DPM and DP-N methods (ignoring DP-T due to its weaker performance on the real data), to the slow drifting data which was used in Fig. 6 of (Buccino and Einevoll 2021) and is available online at https://zenodo.org/record/3696926. Again, detected spikes were projected into PC1-PC2 feature space and DP-N and DPM methods were applied to the frames with the duration of 10 s. For DP-N, we used (10) for mean priors with $K^t = 7$, (11) for mixture probabilities, inverse-Wishart distribution for covariance matrix priors, and two-dimensional Normal distribution for data. The recording length was 60 s and thus, there were 6 individual frames.

As was mentioned in the previous subsection, the output of the clustering in each frame could be used to find templates and then the overlapped spikes which were not detected in the detection step properly and afterward, another clustering phase could be applied by matching the newly detected spikes with the obtained templates.

Figure 8 shows the spikes in the feature space through time frames with colors indicating different clusters. The total error of all clusters in the whole dataset is 6.74% for DP-N, while it is 15.05% for DPM. Calculated error in terms of mean and standard deviation through frames is reported in Table 5. Results clearly show the superior performance of the proposed method, by using the previous frame information for clustering the current interval.

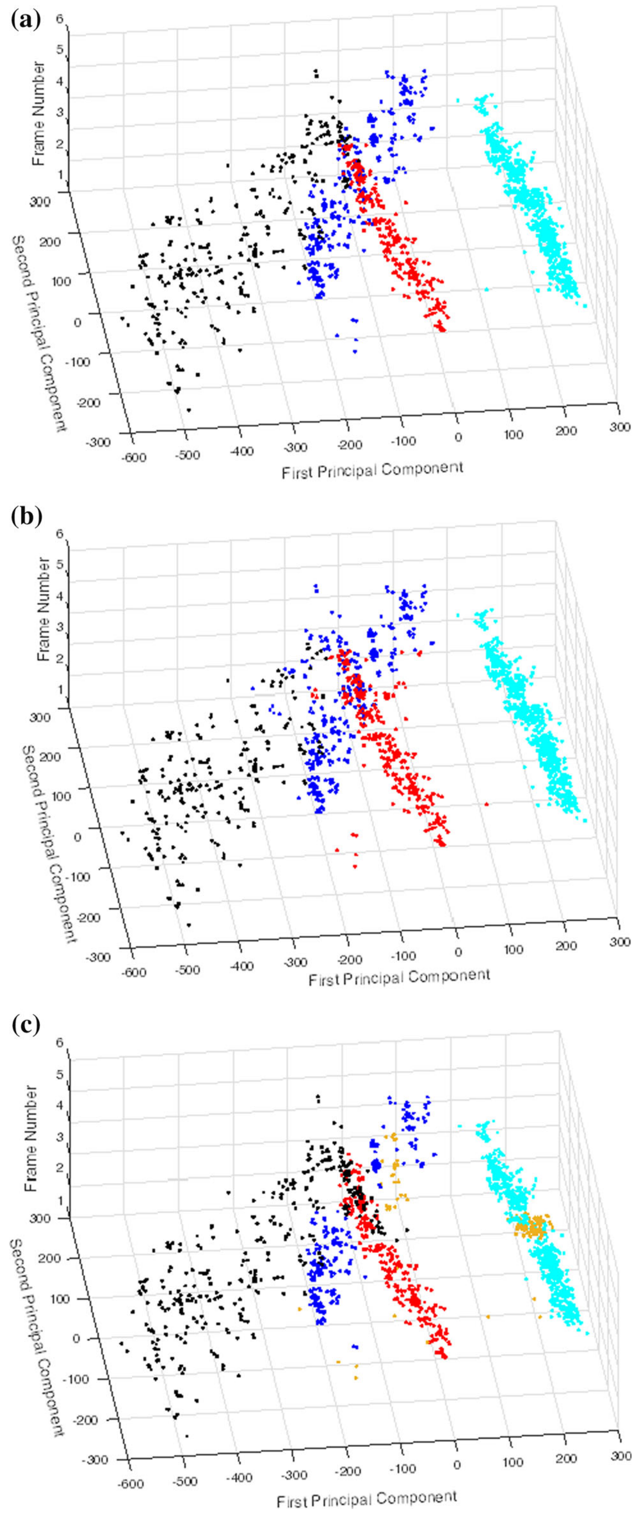

Fig. 8 Spikes in the feature space through time frames, each color indicates an individual cluster which shows the trajectory of clusters through time frames. **a** true clusters, **b** result of DP-N approach, and **c** result of clustering with DPM method

Conclusion and future work

In this article, we presented a Dirichlet process mixture-based method, which is designed to track non-stationary data. This approach uses previous frame information in the current one i.e., in the process of generating cluster means of frame (*t*), cluster means of frame (*t*-1) have been used. By defining the base distribution of means $(\mathbb{G}_0^{\mu})$ in the Dirichlet Process as Normal distribution, tracking Normal drifts of cluster means through time becomes possible. Moreover, because of applying the Dirichlet process mixture, the appearance and disappearance of clusters are also trackable.

We used both t-Student and Normal distributions for data (DP-T and DP-N respectively), and the results show the better performance for Normal distribution. Comparing the results of DP-N and a standard Dirichlet process mixture (which uses no information from the previous frame) shows that in general, the proposed method has better performance in terms of FN, FP, and total error rate as well as appropriate track of cluster means through time. Moreover, the proposed method is simple and does not have the complexities of some other approaches like (Bar-Hillel et al. 2006) and (Wolf and Burdick 2009). For example, without assuming some candidate models with different numbers of components, by using the advantage of the Dirichlet process, the true number of neurons is estimated. In addition, tracking cluster mean Normal drifts is obtained without the assumption that the number of clusters is known and that they are constant through time (like Calabrese and Paninski 2011).

Because of using just the previous frame information in the current one, by applying a software faster than OpenBUGS, real-time implementation is also possible (against Bar-Hillel et al. 2006). In comparison with (Gasthaus et al. 2009), the proposed method does not have the complexities of defining the manner in which model parameters are shared, and in addition, in DP-T and DP-N, Gamma distribution is assumed as Dirichlet process concentration parameter prior, and therefore, its value is obtained related to each frame data.

Despite the so-called advantages, there is the possibility to improve the proposed method and also there are some suggestions for future works. For example, new approaches may be designed with the ability to process data with high speed and to analyze incoming spikes as they are detected rather than waiting for completing a frame. In addition, it would be probable that other feature spaces lead to better results. Moreover, learning features from previous frame spikes along with considering their variations over time can be another suggestion.

Table 5 Error rates of the simulated data

Approaches	Mean of the total error rate	Standard deviation of the total error rate
DP-N	0.0680	0.0667
DPM	0.1559	0.2349

Acknowledgements This work was supported in part by Cognitive Sciences and Technologies Council under Grant No. 1810.

Data Availability The datasets analyzed during the current study are available at http://crcns.org/data-sets/hc/hc-1, and https://zenodo.org/record/3696926.

Declarations

Conflict of interest The authors declare no conflict of interest.

References

Bar-Hillel A, Spiro A, Stark E (2006) Spike sorting: Bayesian clustering of non-stationary data. J Neurosci Methods 157(2):303–316

Brown EN, Kass RE, Mitra PP (2004) Multiple neural spike train data analysis: state-of-the-art and future challenges. Nat Neurosci 7(5):456

Buccino AP, Einevoll GT (2021) Mearec: a fast and customizable testbench simulator for ground-truth extracellular spiking activity. Neuroinformatics 19(1):185–204

Calabrese A, Paninski L (2011) Kalman filter mixture model for spike sorting of non-stationary data. J Neurosci Methods 196(1):159–169

Garcia S, Pouzat C (2019) Tridesclous (software). Github. https://github.com/tridesclous/tridesclous

Gasthaus JA, Wood F (2008) Spike sorting using time-varying DIRICHLET process mixture models. Citeseer

Gasthaus J, Wood F, Gorur D, Teh YW (2009) Dependent Dirichlet process spike sorting. In Advances in neural information processing systems (pp. 497–504)

Gibson S, Judy JW, Markovic D (2010) Technology-aware algorithm design for neural spike detection, feature extraction, and dimensionality reduction. IEEE Trans Neural Syst Rehabil Eng 18(5):469–478

Henze D, Harris K, Borhegyi Z, Csicsvari J, Mamiya A, Hirase H, et al. (2009) Simultaneous intracellular and extracellular recordings from hippocampus region CA1 of anesthetized rats. CRCNS Org

Huang L, Gan L, Ling BWK (2021) A unified optimization model of feature extraction and clustering for spike sorting. IEEE Trans Neural Syst Rehabil Eng 29:750–759

Kamboh AM, Mason AJ (2012) Computationally efficient neural feature extraction for spike sorting in implantable high-density recording systems. IEEE Trans Neural Syst Rehabil Eng 21(1):1–9

Lee J, Mitelut C, Shokri H, Kinsella I, Dethe N, Wu S, Paninski L (2020) YASS: yet Another Spike Sorter applied to large-scale multi-electrode array recordings in primate retina. bioRxiv

Lefebvre B, Yger P, Marre O (2016) Recent progress in multi-electrode spike sorting methods. J Physiol-Paris 110(4):327–335

Lewicki MS (1998) A review of methods for spike sorting: the detection and classification of neural action potentials. Network Comput Neural Syst 9(4):R53-R78

Lunn D, Spiegelhalter D, Thomas A, Best N (2009) The BUGS project: evolution, critique and future directions. Stat Med 28(25):3049–3067

Navratilova Z, Godfrey KB, McNaughton BL (2015) Grids from bands, or bands from grids? An examination of the effects of single unit contamination on grid cell firing fields. Am J Physiol-Heart Circ Physiol

Pachitariu, M., Steinmetz, N., Kadir, S., Carandini, M., & Harris, K. (2016, December). Fast and accurate spike sorting of high-channel count probes with KiloSort. In NIPS proceedings. Neural Information Systems Foundation, Inc..

Paraskevopoulou SE, Barsakcioglu DY, Saberi MR, Eftekhar A, Constandinou TG (2013) Feature extraction using first and second derivative extrema (FSDE) for real-time and hardware-efficient spike sorting. J Neurosci Methods 215(1):29–37

Quiroga RQ (2012) Spike sorting. Curr Biol 22(2):R45–R46

Quiroga RQ, Nadasdy Z, Ben-Shaul Y (2004) Unsupervised spike detection and sorting with wavelets and superparamagnetic clustering. Neural Comput 16(8):1661–1687

Rey HG, Pedreira C, Quiroga RQ (2015) Past, present and future of spike sorting techniques. Brain Res Bull 119:106–117

Sethuraman J (1994) A constructive definition of Dirichlet priors. Statistica Sinica, 639–65.

Shahid S, Walker J, Smith LS (2010) A new spike detection algorithm for extracellular neural recordings. IEEE Trans Biomed Eng 57(4):853–866

Shoham S, Fellows MR, Normann RA (2003) Robust, automatic spike sorting using mixtures of multivariate t-distributions. J Neurosci Methods 127(2):111–122

Soleymankhani A, Shalchyan V (2021) A new spike sorting algorithm based on continuous wavelet transform and investigating its effect on improving neural decoding accuracy. Neuroscience

Teh YW (2011) Dirichlet process. Encyclopedia of machine learning (pp. 280–287), Springer

Wolf MT, Burdick JW (2009) A bayesian clustering method for tracking neural signals over successive intervals. IEEE Trans Biomed Eng 56(11):2649–2659

Wood F, Black MJ (2008) A nonparametric Bayesian alternative to spike sorting. J Neurosci Methods 173(1):1–12

Wood F, Goldwater S, Black MJ (2006) A non-parametric Bayesian approach to spike sorting. In: 2006 international conference of the IEEE engineering in medicine and biology society (pp 1165–1168). IEEE

Wu S-C, Swindlehurst AL (2018) Direct feature extraction from multi-electrode recordings for spike sorting. Digital Signal Process 75:222–231

Yger P, Spampinato GL, Esposito E, Lefebvre B, Deny S, Gardella C, Stimberg M, Jetter F, Zeck G, Picaud S, Duebel J (2018) A spike sorting toolbox for up to thousands of electrodes validated with ground truth recordings in vitro and in vivo. Elife 7:e34518

Yuan Y, Yang C, Si J (2012) The M-Sorter: an automatic and robust spike detection and classification system. J Neurosci Methods 210(2):281–290

Zamani M, Demosthenous A (2014) Feature extraction using extrema sampling of discrete derivatives for spike sorting in implantable upper-limb neural prostheses. IEEE Trans Neural Syst Rehabil Eng 22(4):716–726

Zamani M, Sokolić J, Jiang D, Renna F, Rodrigues MR, Demosthenous A (2020) Accurate, very low computational complexity spike sorting using unsupervised matched subspace learning. IEEE Trans Biomed Circ Syst 14(2):221–231