2014 International Joint Conference on Neural Networks (IJCNN)
July 6-11, 2014, Beijing, China

Neural Signal Analysis by Landmark-based Spectral Clustering with Estimated Number of Clusters

Thanh Nguyen, Abbas Khosravi, Asim Bhatti, Douglas Creighton and Saeid Nahavandi

***Abstract*—Spike sorting plays an important role in analysing electrophysiological data and understanding neural functions. Developing spike sorting methods that are highly accurate and computationally inexpensive is always a challenge in the biomedical engineering practice. This paper proposes an automatic unsupervised spike sorting method using the landmark-based spectral clustering (LSC) method in connection with features extracted by the locality preserving projection (LPP) technique. Gap statistics is employed to evaluate the number of clusters before the LSC can be performed. Experimental results show that LPP spike features are more discriminative than those of the popular wavelet transformation (WT). Accordingly, the proposed method LPP-LSC demonstrates a significant dominance compared to the existing method that is the combination between WT feature extraction and the superparamagnetic clustering. LPP and LSC are both linear algorithms that help reduce computational burden and thus their combination can be applied into real-time spike analysis.**

I. Introduction

Neuroscience practice extracellularly records the activity of single neurons using thin electrodes implanted in the brain. Neurons in the vicinity of the electrode tip are picked up by the extracellular recordings and thus there is a demand to determine which spike corresponds to which neuron. Neurons, which are detected by the same electrode, can react in response to different activities. Even when neighboring neurons have similar responses, it is essential to differentiate them and detect their individual tuning properties, firing characteristics, and connection with other neurons [1]. Spike sorting refers to the process that assigns the detected spikes of a multichannel signal into clusters based on the similarity of their shapes.

In the literature, there exists a number of methods from machine learning or statistical mechanics dealing with neural spike analysis in general or spike sorting in particular. Brown et al. [2] reviewed state-of-the-art techniques and challenges in analysing multiple neural spike training data.

Alternatively, Quiroga [3] proposed a method that combines wavelet transformation (WT) with superparamagnetic clustering without assumptions such as low variance or Gaussian distributions. Hill et al. [4] on the other hand recommend that four quality metrics of false-positive and false-negative errors should accompany spike sorting regardless of the algorithm used to sort. These metrics would facilitate the assessment regarding the performance of the sorter relative to the level of contamination of the data.

Oliynyk et al. [5] constructed a tool for quick and robust online classification of single neuron activity using the fuzzy c-means clustering. The method is particularly suitable for the analysis of large parallel recordings, which are practically impossible or inconvenient for human supervision, and thus is helpful in the decoding of neural ensembles or other clinical applications.

Shalchyan et al. [6] introduced an algorithm for automatic unsupervised detection of action potentials in extracellular recordings. A new manifestation variable for detection is defined based on the combination of denoised wavelet coefficients over selected scales. Tiganj and Mboup [7] used an iterative application of independent component analysis along with a deflation technique in two nested loops for neural signal analysis with multi-channel recordings. Each loop of the algorithm enhances the final sorting results and thus significantly improves the overall spike sorting performance.

More recently, Shao et al. [8] examined spike sorting errors and their impacts to the Granger causality analysis, which is a powerful technique for detecting causal interactions between time series signals. Pillow et al. [9] otherwise investigated the geometry of failures of traditional spike sorting algorithms and developed a sorting model, which explicitly takes into account the superposition of spike waveforms.

Though various methods have been proposed, obtaining high accuracy in spike sorting is always a big challenge in neuroscience and biomedical engineering. Furthermore, the computational burden in spike sorting is massive. This paper presents an integrated approach, which combines locality preserving projection (LPP) [10], gap statistics (GS) [11], and landmark-based spectral clustering (LSC) [12], for a computationally inexpensive unsupervised spike sorting method. According to our best knowledge, this is the first proposal on application of the LPP for spike feature extraction, GS in determining the number of clusters, and LSC for clustering spike sorting data. The accuracy of the proposed approach is compared to the renowned benchmark spike sorting method that is a combination between the wavelet transformation (WT) and the superparamagnetic clustering (SPC) in [3].

The authors are with Centre for Intelligent Systems Research (CISR), Deakin University, Geelong Waurn Ponds Campus, VIC, Australia (phone: +61352278281; fax: +61352271046; e-mail: thanh.nguyen@deakin.edu.au)

978-1-4799-1484-5/14/$31.00 ©2014 IEEE

The arguments are organized as follows. The next section describes details of steps in the proposed methodology. Section III is devoted for experiments and results whilst discussions and concluding remarks are presented in Section IV & V respectively.

II. Spike Sorting Methodology

The proposed methodology is graphically illustrated in Fig. 1 where LPP method is employed for spike feature extraction. The automatic unsupervised clustering is deployed by a combination of GS and LSC.

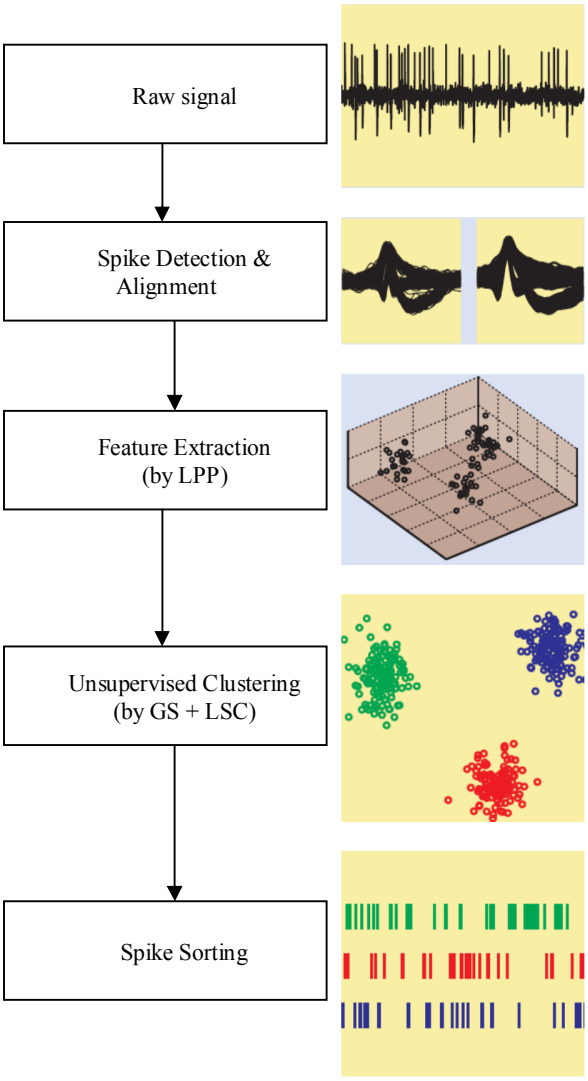

Fig. 1. Spike sorting proposed method (adapted from [13])

The first step in the methodology is spike detection, which aims to identify data points that form an action potential. The voltage threshold detection is employed where the automatic threshold (Thr) is set to:

$$Thr = 4median\left\{\frac{|x|}{0.6745}\right\}$$

where x is the bandpass-filtered signal [3]. For each detected spike, 64 samples are assembled for further process. Details of other steps are described in the following subsections.

A. Feature Extractions

Feature extraction is one of the most important steps in which the silent features of the spikes are derived based on spike wave shapes. The features should be able to well differentiate spikes of different neurons and preferably low-dimensional. Simple features like peak-to-peak amplitude, maximum spike amplitude and spike width can be used [13]. These approaches however are sensitive to noise and intrinsic variations in spike shapes. Alternatively, principal component analysis (PCA) is one of the popular methods used for feature extraction [14-16]. WT also has emerged as a competitive feature extraction method for spike sorting [17-20]. For ease of comparison, we briefly present both WT and the suggested method LPP in the following subsections.

1) Wavelet Transformation (WT): WT represents a signal in a time-frequency fashion [21]. Once the wavelets (the mother wavelet) $\varphi(x)$ is fixed, translations and dilations of the mother wavelet can be formed $\left\{\varphi\left(\frac{x-b}{a}\right), (a,b) \in \mathrm{R}^{+} \times \mathrm{R}\right\}$. It is convenient to take special values for a and b as $a = 2^{-j}$ and $b = 2^{-j}k$ where j and k are integers. One of the simplest wavelets is the Haar wavelet, which has been used in various applied mathematics. Haar functions can uniformly approximate any continuous function. Dilations and translations of the function φ, which is $\varphi_{jk}(x) = const.\varphi(2^{j}x - k)$, define an orthogonal basis in $L^2(R)$. This means that any element in $L^2(R)$ may be represented as a linear combination of these basis functions. The scaling function in Haar wavelet is simply unity on the interval [0,1) as $\phi(x) = 1\ (0 \le x < 1)$. Quiroga et al. [3] employed a four-level decomposition based on Haar wavelets for spike sorting. The wavelet coefficients are then selected by the normality test based on the Lilliefors modification of the Kolmogorov-Smirnov (KS) test.

2) Locality Preserving Projections (LPP): LPP is a linear algorithm, which is seen as an alternative to PCA [10]. It constructs a graph integrating neighbourhood information of the dataset. A transformation matrix mapping the data points to a subspace is computed based on the notion of the Laplacian of the graph. LPP obtains a subspace spanned by the smallest eigenvectors of the local covariance matrix. Three steps of LPP are summarized below:

Step 1. Creating the adjacency graph: Define G as a graph with m nodes. If x_i and x_j are close, an edge between nodes i and j are created. There are two alternatives:

(a) ϵ-neighborhoods: Nodes i and j are linked by an edge if $\|x_i - x_j\|^2 < \epsilon$ where $\epsilon \in R$ and the norm is the Euclidean norm in R^n.

(b) k nearest neighbors, $k \in N$. Nodes i and j are linked by an edge if i is among k nearest neighbours of j or j is amongst k nearest neighbours of i.

Step 2. Choosing the weights. There are two variations for weighting the edges as well. W is a sparse symmetric m-dimensional matrix with W_{ij} having the weight of the edge connecting vertices i and j, and 0 if there is no connection between i and j.

(a) Heat kernel. If nodes i and j are linked, where $t \in R$ set $W_{ij} = e^{-\frac{\|x_i - x_j\|^2}{t}}$;

(b) Simple-minded. $W_{ij} = 1$ if and only if vertices i and j are linked by an edge.

Step 3. Eigenmaps: Calculate the eigenvectors and eigenvalues for the generalized eigenvector problem:

$$XLX^T \boldsymbol{a} = \lambda XDX^T \boldsymbol{a}$$

where D is a diagonal matrix with entries are column (or row if W is symmetric) sums of W, $D_{ij} = \sum_j W_{ji}$ and $L = D - W$ is the Laplacian matrix. Note that the ith column of matrix X is x_i.

Denote $[a_0, \dots, a_{l-1}]^T$ as the solution of the above problem, ordered based on their eigenvalues, $\lambda_0 < \cdots < \lambda_{l-1}$. Thus the embedding is characterized as follows:

$$x_i \to y_i = A^T x_i, A = (a_0, \dots, a_{l-1})$$

where y_i is a l-dimensional vector, and A is an $n \times l$ matrix.

In this paper, instead of using WT, we suggest the linear algorithm LPP for extracting spike features.

B. Clustering Methods

Three clustering methods are investigated in this paper. Along with SPC and LSC, the well-known k-means clustering is also examined for extra comparisons.

1) Superparamagnetic clustering (SPC): SPC is one of the unsupervised clustering methods based on the simulated connections between each data sample and its K nearest neighbours [22]. SPC calculates the interaction strength J_{ij} between neighboring points v_i and v_j in the first step:

$$J_{ij} = \begin{cases} \frac{1}{K} exp\left(-\frac{d_{ij}^2}{2a^2}\right), & \text{if } v_i \text{ and } v_j \text{ are neighbors} \\ 0, & \text{otherwise} \end{cases}$$

where a and K are the average distance and the number of nearest neighbors. The strength of the interaction J_{ij} decreases exponentially if the Euclidean distance $d_{ij} = \|x_i - x_j\|^2$ increases. The Euclidean distance simply represents the similarity of spikes. Therefore, points in high density areas have stronger connections than those in low density areas. In the next step, each point x_i is assigned an initial random state s from 1 to q. Then N Monte Carlo iterations are carried out for various temperatures T employing the Wolf algorithm. Given an initial structure of states s, a point x_i is randomly chosen and its state s altered to a new state s_{new} somewhere between 1 and q. The likelihood that the nearest neighbors of x_i will also change their state to s_{new} is computed as:

$$p_{ij} = 1 - exp\left(-\frac{J_{ij}}{T} \delta_{s_i, s_j}\right)$$

where T is the temperature and δ_{s_i, s_j} is the point-point correlation [3].

In SPC, the temperature has a key role in determining the number of clusters. All points will change their state together at low temperatures and therefore will be considered a single cluster. In contrast, at high temperatures, many points will change their state independently and thus partitioning the data into several clusters. The challenge is that one needs to pre-specify a range of temperature, e.g. $T = [0,0.2]$ in [3].

There is no doubt that this range of temperature may be selected imprecisely because different datasets with different noise levels would be in favour of different temperatures. Furthermore, selecting a right temperature in the range may involve mistakenness due to the existence of unlimited points in the range. Even if the range is pre-defined appropriately, the algorithm to select the correct temperature to result in correct number of clusters is not guaranteed to work precisely in every circumstance.

2) K-means Clustering: K-means clustering [23] partitions N observations of d-dimension into K clusters in which each observation assigned to the cluster with the nearest mean. Given a set of observations $(x_1, x_2, \dots, x_N)$, k-means clustering partitions the N observations into K clusters $(K \leq N)$ $C = \{C_1, C_2, \dots, C_K\}$ with the aim to minimize the within-cluster sum of squares. The algorithm comprises a simple re-estimation process including the following two steps.

Step 1: The data points are allocated randomly to the K clusters and the centroid is calculated for each set:

$$m_k = \frac{\sum_{i:C(i)=k} x_i}{N_k}, \qquad k = 1, \dots, K$$

where N_k and m_k is the number of data points and centre of the kth cluster respectively.

Step 2: Every point is allocated to the cluster whose centroid is closest to that point using the squared Euclidean distance measure. For a current set of cluster means, assign each observation as:

$$C(i) = \underset{1 \leq k \leq K}{\operatorname{argmin}} \|x_i - m_k\|^2, \quad i = 1, \dots, N$$

These two steps are repeated and halted when a stopping condition is met, i.e. there is no further change in the allocation of the data points.

3) Landmark-based Spectral Clustering (LSC): LSC was first introduced by Chen and Cai [12]. The basic idea of LSC is to design an efficient way for graph construction and Laplacian matrix eigen-decomposition. LSC initially selects p representative data points as the landmarks and characterize the remaining observations as the linear combinations of these landmarks. The spectral embedding of the data can then be effectively calculated with the landmark-based representation. This proposed procedure scales linearly with the problem size. LSC is briefed in the following five steps.

Step 1. Create p landmark points based on k-means or random selection.

Step 2. Produce a sparse affinity matrix $Z \in R^{p\times n}$ between observations and landmark points, with the affinity computed using the following equation.

$$z_{ji} = \frac{K_h(x_i, u_j)}{\sum_{j' \in U_{\langle i \rangle}} K_h(x_i, u_{j'})}$$

where $K_h(.)$ is a kernel function with a bandwidths h and $j \in U_{\langle i \rangle}$ with $U_{\langle i \rangle} \in R^{m\times r}$ denote a sub-matrix of U consisted of r nearest landmarks of x_i.

The Gaussian kernel $K_h(x_i, u_j) = \exp(-\|x_i - u_j\|^2/2h^2)$ is one of the most commonly used.

Step 3. Calculate the first k eigenvectors of ZZ^T, denoted by $A = [a_1, \dots, a_k]$.

Step 4. Calculate $B = [b_1, \dots, b_k] \in R^{n\times p}$ by $B^T = \sum^{-1} A^T \hat{Z}$ where $\hat{Z} = D^{-1/2}Z$ with D is the row-sum of Z.

Step 5. Apply k-means clustering to the matrix B to get the output clusters.

In experiments of this paper, the k-means is chose for selecting landmark points with $p = 1000$. The number of nearest landmarks r is set equal to 5 whilst the number of cluster k must be found in advance by GS, which is presented in the next subsection.

C. Evaluating Clusters: Gap Statistics (GS)

In k-means or LSC, the challenge is to find the number of clusters before performing the clustering. The following presents a popular cluster evaluator GS [11] to find the natural number of clusters as an input of k-mean or LSC. GS has been found successfully applied into a wide range of problems, especially in biomedical engineering [24-25]. It is thus motivated for using in this particular spike sorting problem. GS procedure employed herein comprises the following nine steps [26]:

Step 1. Cluster the data using the desired clustering method to obtain partitions $k = 1,2, \dots, K$.

Step 2. Calculate the observed $\log(W_k)$ for each partition with K clusters.

Step 3. Create a random sample X^* of size n using the gap-uniform procedure. For each of the i dimensions (or variables), we generate n one-dimensional variates that are uniformly distributed over the range x_i^{min} to x_i^{max}, where x_i characterizes the i-th variable or the i-th column of X.

Step 4. Cluster the random sample X^* by the same clustering technique as in Step 1).

Step 5. Calculate the within-dispersion measures $\log(W_{k,b}^*)$ for this sample.

Step 6. Repeat steps 3) and 5) for B times to result in a set of measure $\log(W_{k,b}^*)$ where $k = 1, \dots, K$ and $b = 1, \dots, B$.

Step 7. Compute the average of these values and their standard deviation as follows:

$$\overline{\mathrm{W}_k} = \frac{1}{B}\sum_b \log(W_{k,b}^*)$$

$$sd_k = \sqrt{\frac{1}{B}\sum_b \left[\log(W_{k,b}^*) - \overline{\mathrm{W}}\right]^2}$$

Step 8. Calculate the estimated gap statistics

$$\mathrm{gap}(k) = \overline{\mathrm{W}_k} - \log(\mathrm{W}_k)$$

Step 9. Define $\mathrm{s}_k = sd_k\sqrt{1 + 1/B}$ and chose the number of clusters as the smallest k such that $\mathrm{gap}(k) \geq \mathrm{gap}(k+1) - s_{k+1}$.

It is worth to note that the challenge to obtain the likely correct number of clusters is as difficult as obtaining the optimal temperature in SPC. Selecting a number of clusters far from being optimal can cause a dramatic change in accuracy of the model.

III. EXPERIMENTS AND RESULTS

Datasets used for experiments in this paper are identical to those published in Quiroga et al. [3] for ease of comparisons. Four datasets with different noise levels, from 0.05 to 0.2 (or 0.4 for easy clustering datasets), are simulated for experiments. Three separate spikes having a Poisson distribution of interspike intervals with a mean firing rate of 20 Hz are presented in all four datasets. The background noise reproduces spike shape variability although spikes are originated from the same class. The refractory period of 2ms is introduced between spikes of the same class.

In the previous study Quiroga et al. [3], the WT technique is used for feature extraction to derive 64 features of the spike shapes. These features are then reduced using the KS test for normality. The first 10 wavelet coefficients with the largest deviation from normality were employed for implementation. The chosen wavelet coefficients, which are a compressed form of the spike features, are then input to the SPC clustering method. Experimental results based on the synthetic data demonstrated that the combination of WT (for feature extraction) with SPC leads to higher spike sorting accuracy compared to other combinations, i.e. PCA + SPC, Wavelet + K-means or PCA + K-means (the first term represents feature extraction whilst the latter is the clustering method).

Comparable to 10 wavelet coefficients, 10 features derived from the LPP method are served as inputs to clustering methods. Graphical comparisons of feature extraction by wavelet plus KS test and by the LPP method are presented in Fig. 2 & 3 respectively. The dataset at Example 1, noise level 0.2 (see Table I) is used for the illustration. Note that results reported here are with and without overlapping spikes after clustering by the k-mean method.

Comparing the projections before and after removing overlapping spikes in both methods, it is clear that overlapping spikes do not belong to a cluster clearly but they tend to locate in the confused areas among clusters. Obviously, overlapping spikes are hard to be clustered and they certainly can cause a dramatic decline in the performance of the feature extraction and then of the clustering methods.

Both feature extraction methods WT and LPP demonstrate the ability to separate three clusters. However, it is worth to note the appearance of many red colour spikes in the blue cluster area in the projection with overlapping spikes (Fig.

2A) and even after removing overlapping spikes (Fig. 2B) of the WT method. In the LPP projection, after removing overlapping spikes, the mixture among 3 clustering areas disappear and fetch very clear separate clusters (Fig. 3B). Thus, very discriminative features are obtained by the LPP compared to the WT+KS method.

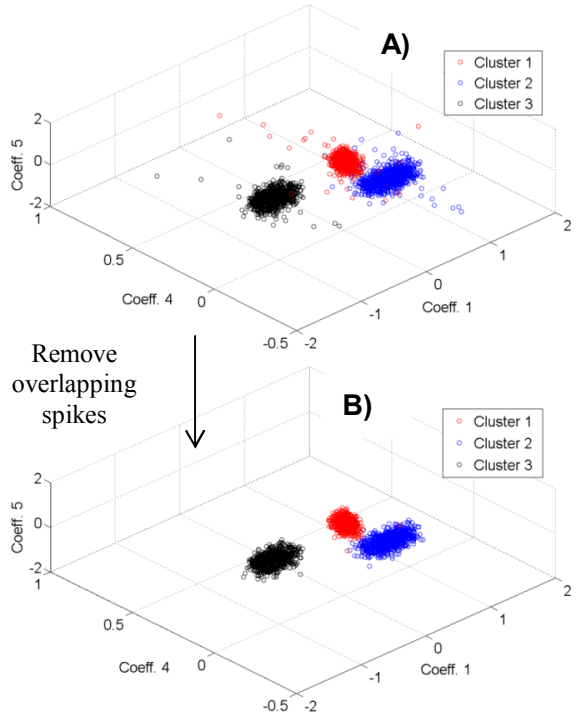

Fig 2. Best projection of extracted features by WT + KS test: A) with overlapping spikes, B) without overlapping spikes.

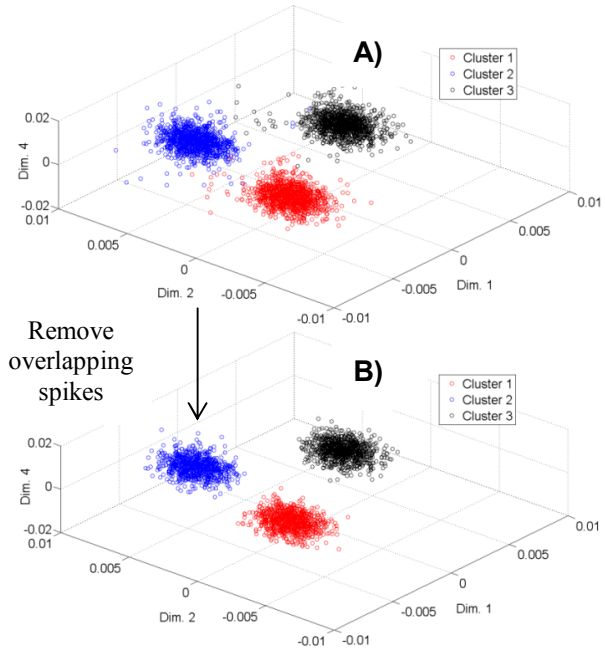

Fig 3. Best projection of extracted features by LPP: A) with overlapping spikes, B) without overlapping spikes.

After extracting features of the spike waveforms by either WT or LPP, each clustering algorithm, i.e. SPC, k-means, and LSC, is carried out 20 times and the average results are reported in Table I & II.

Table I & II present results in the fashion that separates two feature extractions: WT and LPP. Table I assembles results when having overlapping spikes included whilst outcomes in Table II are calculated without overlapping spikes. For each dataset, results presented in Table I & II are of independent trails.

Note that in SPC, there is not a pre-set number of clusters but SPC finds the number of clusters automatically. SPC selects the appropriate temperature that decisively determines the number of clusters. As SPC does not require a prior determination of the number of clusters, the threshold of 3 clusters (equal to the real number of clusters) is set to limit the number of clusters. If the number of clusters automatically detected is greater than 3, then the 3 largest clusters with the most overlapping with the real clusters will be recorded to calculate the accuracy.

On the other hand, the k-means and LSC methods require a pre-determined number of clusters by gap statistics. Therefore, the numbers in parentheses adjacent to values in the k-means and LSC columns indicate the number of clusters when different from 3.

IV. Discussions

Comparing two feature extraction methods, we see that spike features extracted by WT and LPP when inputting into the SPC lead to a very similar accuracy. With overlapping spikes (Table I), WT+SPC obtains 65.35% accuracy whilst LPP+SPC reaches 65.41%. Without overlapping spikes, these numbers are 68.20% and 68.96% respectively in Table II. However, when applying to the k-means and LSC methods, there is a significant difference. LPP features result in a much greater accuracy than those of the WT in both Table I and Table II. Specifically, LPP combined with k-means on average obtains the accuracy at 74.37% (or 74.63% in Table II) whilst that of the WT+K-means is just at 68.22% (or 68.60% in Table II). The proposed method LPP+LSC leads to the highest average accuracy, at 78.20% (or 81.63% in Table II) compared to 73.65% (or 75.04% in Table II) of the WT+LSC.

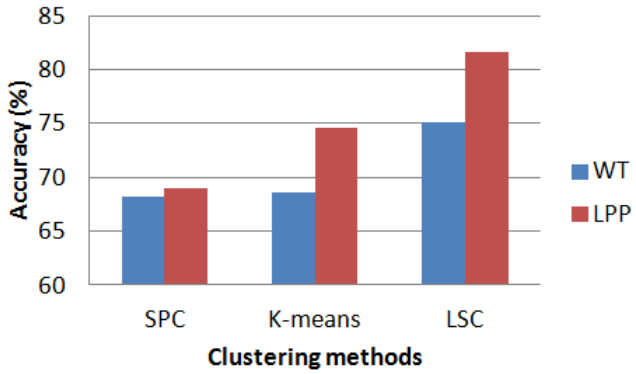

Fig. 4. Average performance of three clustering methods

TABLE I
AVERAGE ACCURACY OF DIFFERENT SPIKE SORTING METHODS

Example	Noise levels	No. spikes	WT SPC(*)	WT GS + K-means	WT GS + LSC	LPP SPC(*)	LPP GS + K-means	LPP GS + LSC
Example 1	0.05	3514	58.56	77.96 (5)	83.89 (4)	54.83	62.85 (5)	61.35 (5)
	0.10	3522	62.68	77.24 (5)	68.93 (5)	62.29	73.21 (5)	68.14 (5)
	0.15	3477	71.19	72.81 (5)	69.75 (5)	77.57	72.25 (5)	69.08 (5)
	0.20	3474	60.52	69.57 (5)	83.17 (4)	84.07	83.06 (4)	98.73
	0.25	3298	70.73	70.77 (5)	69.09 (5)	79.62	90.89	99.18
	0.30	3475	50.88	79.66 (4)	67.36 (5)	55.23	87.59	84.05 (4)
	0.35	3534	69.58	68.34 (5)	67.56 (5)	56.08	80.93	83.46 (4)
	0.40	3386	87.21	67.69 (5)	66.03 (5)	54.28	88.00	99.38
Example 2	0.05	3410	67.79	73.37 (5)	65.12 (5)	63.90	66.92 (5)	75.01 (5)
	0.10	3520	81.10	60.10 (5)	99.07	79.44	80.64 (5)	78.30 (5)
	0.15	3411	70.74	59.80 (4)	67.67 (5)	75.08	85.01 (4)	83.14 (4)
	0.20	3526	60.13	50.55 (5)	66.90 (5)	42.54	89.81	65.93 (2)
Example 3	0.05	3383	63.17	73.71 (5)	50.99 (5)	53.59	48.21 (2)	55.70 (5)
	0.10	3448	53.64	51.59 (5)	67.78 (5)	67.57	68.55 (5)	65.09 (5)
	0.15	3472	66.26	47.40	97.80	38.31	44.91 (2)	65.46 (2)
	0.20	3414	52.86	47.89 (5)	63.50 (5)	34.68	40.62 (2)	59.18 (2)
Example 4	0.05	3364	58.69	73.23 (5)	67.39 (5)	64.23	81.04 (5)	90.34 (5)
	0.10	3462	55.36	73.42 (5)	82.97 (4)	88.91	58.52 (2)	65.56 (2)
	0.15	3440	68.11	96.96	68.30 (5)	89.68	93.64	97.66
	0.20	3493	77.75	72.36 (5)	99.67	86.32	90.80	99.31
Average		**3451**	**65.35**	**68.22**	**73.65**	**65.41**	**74.37**	**78.20**

(*)No pre-set number of clusters for SPC

TABLE II
AVERAGE ACCURACY OF DIFFERENT SPIKE SORTING METHODS WITHOUT OVERLAPPING SPIKES

Example	Noise levels	No. spikes	WT SPC(*)	WT GS + K-means	WT GS + LSC	LPP SPC(*)	LPP GS + K-means	LPP GS + LSC
Example 1	0.05	2729	60.63	78.97 (5)	85.37 (4)	54.72	54.06 (5)	73.63 (5)
	0.10	2753	61.80	77.57 (5)	67.74 (5)	62.06	70.20 (5)	76.37 (5)
	0.15	2693	72.29	74.52 (5)	69.39 (5)	76.25	68.98 (5)	68.32 (5)
	0.20	2678	66.26	71.81 (5)	83.95 (4)	89.39	85.61 (4)	99.95
	0.25	2586	69.80	75.37 (5)	69.39 (5)	84.32	87.68	99.98
	0.30	2629	47.82	83.61 (4)	67.38 (5)	58.74	90.80	84.36 (4)
	0.35	2702	68.16	65.96 (5)	67.85 (5)	56.61	72.33	84.38 (4)
	0.40	2645	93.85	69.81 (5)	67.31 (5)	54.64	87.89	99.76
Example 2	0.05	2619	66.74	74.99 (5)	68.98 (5)	67.59	74.23 (5)	83.29 (5)
	0.10	2694	80.90	63.18 (5)	99.93	82.70	87.91 (5)	84.98 (5)
	0.15	2648	78.70	58.13 (4)	62.66 (5)	85.86	85.76 (4)	84.92 (4)
	0.20	2715	67.71	51.84 (5)	67.33 (5)	44.82	82.29	66.53 (2)
Example 3	0.05	2616	75.67	77.39 (5)	69.88 (5)	57.04	55.33 (2)	67.82 (5)
	0.10	2638	63.44	42.23 (5)	69.20 (5)	70.07	68.68 (5)	62.89 (5)
	0.15	2660	60.64	46.97	99.88	40.12	48.60 (2)	67.29 (2)
	0.20	2624	51.45	48.64 (5)	66.94 (5)	34.72	42.70 (2)	60.52 (2)
Example 4	0.05	2535	61.02	75.88 (5)	67.72 (5)	72.86	89.13 (5)	100.0 (5)
	0.10	2742	62.26	73.19 (5)	82.89 (4)	99.21	58.31 (2)	67.65 (2)
	0.15	2631	74.85	87.85	67.08 (5)	97.70	85.22	99.99
	0.20	2716	80.08	74.09 (5)	100.0	89.73	96.87	99.99
Average		**2663**	**68.20**	**68.60**	**75.04**	**68.96**	**74.63**	**81.63**

Comparing three clustering methods, LSC on average is clearly more accurate than k-means and SPC (see Fig. 4 for a graphical comparison). Along with the superiority of LPP features against WT ones, it is obvious that the proposed LPP+LSC is the more competent spike sorting compared to the popular benchmark WT+SPC. LPP+LSC is approximately on average 13% superior to WT+SPC regardless of taking or not taking overlapping spikes into account (see Table I & II). The box plot below exhibits more detailed the dominance of LPP+LSC contrast to WT+SPC.

Fig. 5 illustrates the comparisons between the proposed method LPP+LSC (in light blue) and the benchmark WT+SPC (in red colour) after removing overlapping spikes. In each dataset, denoted by its noise level on the horizontal axis, there are two boxes showing results of two corresponding methods. Each box shows the distribution of results throughout 20 trials.

The median values of the LPP+LSC distributions are greater than those of the WT+SPC in almost every experimental dataset. This is consistent with the mean values reported in Table II that shows a significant dominance of the proposed LPP+LSC method against the WT+SPC. Alternatively, the interquartile ranges of the LPP+LSC are much smaller than those of the WT+SPC. This demonstrates the greater stability and robustness of the LPP+LSC approach compared to WT+SPC.

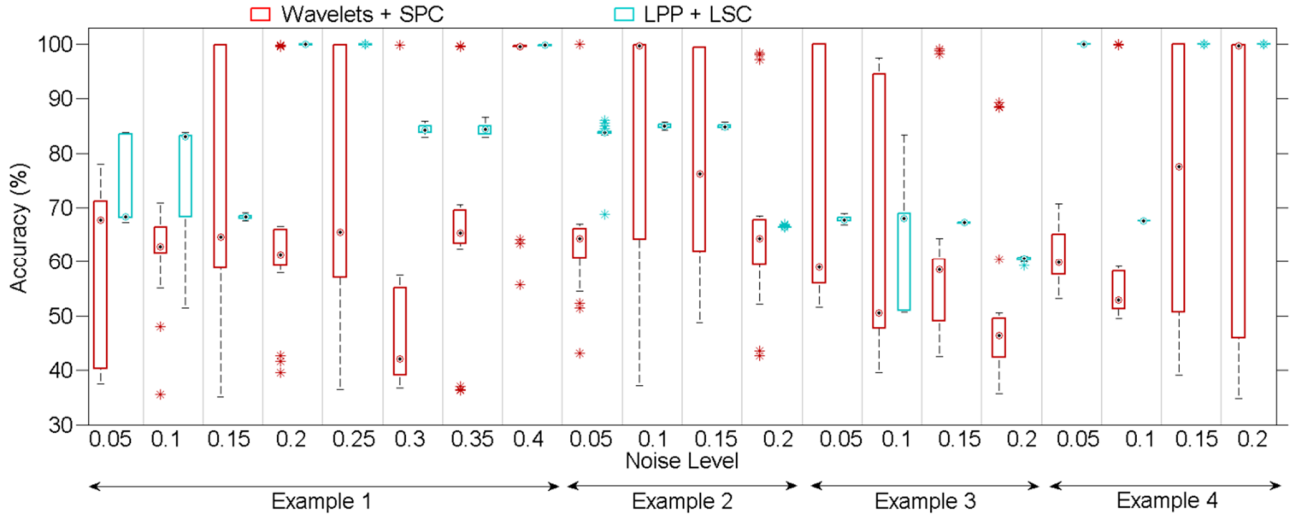

Fig. 5. Performance comparisons between WT+SPC and LPP+LSC methods

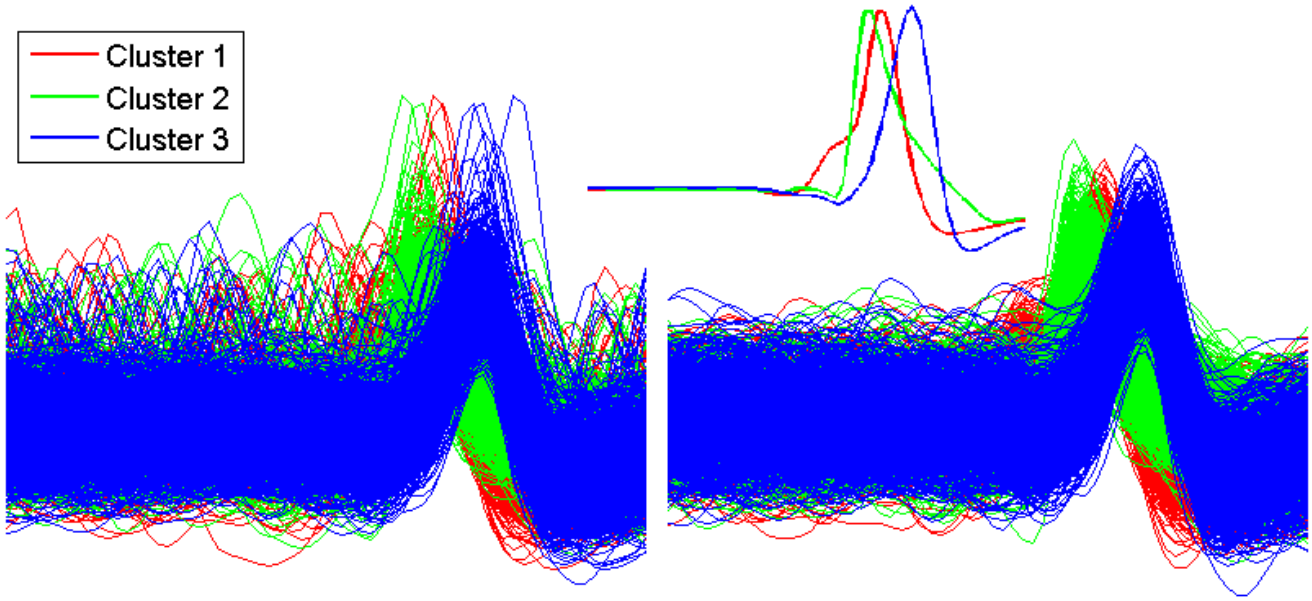

Fig. 6. Spikes after clustering of Example 4, noise level 0.2

On the other hand, it is seen that GS when deployed with LPP features obviously leads to a more accurate number of clusters than when applied with WT features.

The number of clusters detected reflects not only the performance of GS but also the effectiveness of the feature extraction method. LPP features enable GS to perform more efficiently than those of the WT. Particularly, when GS combined with k-means, LPP leads to seven correct cases (3 clusters) whilst WT produces correct results in only two datasets. In the case of LSC, LPP leads to precise results in five datasets compared to three datasets of the WT features (see Table I or II).

Fig. 6 graphically shows results of the proposed method applied to the dataset Example 4, noise level 0.2. In the figure, the left presents both overlapping and none-overlapping spikes whilst the right has overlapping ones removed. The original spike shapes of the dataset are plotted on top of the figure. It is seen that the spikes in the dataset exhibit a complicated spike sorting process. This is because the shapes of its 3 spikes are hard to differentiate (see original spikes on top of Fig. 6).

V. Conclusions

This paper presents the integrated approach of LPP, GS and LSC for spike sorting. As per obtained results, the proposed method significantly dominates the popular existing combination of WT and SPC. LPP method exhibits its remarkable suitability in the spike feature extraction compared to the WT.

On the other hand, the LSC clustering method demonstrates its advantage when combined with the cluster evaluator GS. Under the automatic perspective, the

comparison is absolutely convincing as the challenge to find the correct number of clusters in the proposed LPP+LSC method is as difficult as to find the appropriate temperature in SPC. Selecting an inappropriate temperate in SPC results in a low spike sorting accuracy and so does the cluster evaluation by GS. The challenge to estimate the correct range of temperature for SPC is complicated because datasets with different noise levels would require different temperatures.

Given that the range of temperature is specified correctly, selecting the ideal single point temperature from the range is also problematic for SPC due to the existence of unlimited points within the range. Our proposed method attempts to estimate the number of clusters first and then employs clustering methods that lead to significantly improved spike sorting performance.

Overlapping spikes are a huge challenge for any spike sorting methods as they tend to increase the difficulties in detection and sorting and thus reduce the overall sorting performance. Overlapping spike sorting problem thus would be worth a further research. As LPP and LSC are both linear algorithms, the combination of these two methods helps reduce remarkably the computational burden. This implies that the proposed method can be applied into real-time spike analysis, which is interesting for a more extensive investigation in the future.

REFERENCES

[1] R.Q. Quiroga, "Spike sorting," *Current Biology*, vol. 22, no. 2, pp. R45-R46, 2012.

[2] E.N. Brown, R.E. Kass, & P.P. Mitra, "Multiple neural spike train data analysis: state-of-the-art and future challenges," *Nature Neuroscience*, 7(5), 456-461, 2004.

[3] R.Q. Quiroga, Z. Nadasdy, & Y. Ben-Shaul, "Unsupervised spike detection and sorting with wavelets and superparamagnetic clustering," *Neural Computation*, 16(8), 1661-1687, 2004.

[4] D.N. Hill, S.B. Mehta, & D. Kleinfeld, "Quality metrics to accompany spike sorting of extracellular signals," *Journal of Neuroscience*, 31(24), 8699-8705, 2011.

[5] A. Oliynyk, C. Bonifazzi, F. Montani, & L. Fadiga, "Automatic online spike sorting with singular value decomposition and fuzzy c-mean clustering," *BMC Neuroscience*, 13(1), 96, 2012.

[6] V. Shalchyan, W. Jensen, & D. Farina, "Spike detection and clustering with unsupervised wavelet optimization in extracellular neural recordings," *IEEE Transactions on Biomedical Engineering*, 59(9), 2576-2585, 2012.

[7] Z. Tiganj, & M. Mboup, "Neural spike sorting using iterative ICA and a deflation-based approach," *Journal of Neural Engineering*, 9(6), 066002, 2012.

[8] P.C. Shao, W.T. Tseng, C.C. Kuo, W.C. Shann, M.L. Tsai, & C.C. Yen, "Effects of spike sorting error on the Granger causality index," *Neural Networks*, 46, 249-259, 2013.

[9] J.W. Pillow, J. Shlens, E.J. Chichilnisky, & E.P. Simoncelli, "A model-based spike sorting algorithm for removing correlation artifacts in multi-neuron recordings," *PloS One*, 8(5), e62123, 2013.

[10] X. He and P. Niyogi. Locality preserving projections. *Advances in Neural Information Processing Systems*, vol. 16, page 37, Cambridge, MA, USA, The MIT Press, 2004.

[11] R. Tibshirani, G. Walther, & T. Hastie, "Estimating the number of clusters in a data set via the gap statistic," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(2), 411-423, 2001.

[12] X. Chen & D. Cai, "Large scale spectral clustering with landmark-based representation," *Proc. Twenty-Fifth AAAI Conf. Artificial Intelligence*, pp. 313-318, 2011.

[13] S. Gibson, J.W. Judy, & D. Markovic, "Spike sorting: The first step in decoding the brain," *IEEE Signal Processing Magazine*, 29(1), 124-143, 2012.

[14] H.K. Jung, J.H. Choi, & T. Kim, "Solving alignment problems in neural spike sorting using frequency domain PCA," *Neurocomputing*, 69(7), 975-978, 2006.

[15] Z. Tiganj & M. Mboup, "A non-parametric method for automatic neural spike clustering based on the non-uniform distribution of the data," *Journal of Neural Engineering*, 8(6), 066014, 2011.

[16] J. Wild, Z. Prekopcsak, T. Sieger, D. Novak, & R. Jech, "Performance comparison of extracellular spike sorting algorithms for single-channel recordings," *Journal of Neuroscience Methods*, 203(2), 369-376, 2012.

[17] H.L. Chan, T. Wu, S.T. Lee, M.A. Lin, S.M. He, P.K. Chao, & Y.T. Tsai, "Unsupervised wavelet-based spike sorting with dynamic codebook searching and replenishment," *Neurocomputing*, 73(7), 1513-1527, 2010.

[18] C. Yang, B. Olson, & J. Si, "A multiscale correlation of wavelet coefficients approach to spike detection," *Neural Computation*, 23(1), 215-250, 2011.

[19] H.Y. Lai, Y.Y. Chen, S.H. Lin, Y.C. Lo, S. Tsang, S.Y. Chen,... & F.S. Jaw, "Automatic spike sorting for extracellular electrophysiological recording using unsupervised single linkage clustering based on grey relational analysis," *Journal of Neural Engineering*, 8(3), 036003, 2011.

[20] Y. Yuan, C. Yang, & J. Si, "The M-Sorter: An automatic and robust spike detection and classification system," *Journal of Neuroscience Methods*, 210(2), 281-281, 2012.

[21] R.A. DeVore & B.J. Lucier, "Wavelets," *Acta Numerica*, 1(1), 1-56, 1992.

[22] M. Blatt, S. Wiseman, & E. Domany, "Data clustering using a model granular magnet,' *Neural Computation*, 9(8), 1805-1842, 1997.

[23] S. Lloyd, "Least squares quantization in PCM," *IEEE Transactions on Information Theory*, 28(2), 129-137, 1982.

[24] M.C. Wendl & S.P. Yang, "Gap statistics for whole genome shotgun DNA sequencing projects," *Bioinformatics*, 20(10), 1527-1534, 2004.

[25] Z. Yin, X. Zhou, C. Bakal, F. Li, Y. Sun, N. Perrimon, & S.T. Wong, "Using iterative cluster merging with improved gap statistics to perform online phenotype discovery in the context of high-throughput RNAi screens," *BMC Bioinformatics*, 9(1), 264, 2008.

[26] W.L. Martinez and A.R. Martinez, "Finding clusters," In *Exploratory Data Analysis with MATLAB*, Chapman and Hall/CRC, 2004.