Spike Sorting Based on Window-Gradient Feature

Min Dai1,*, Feng Jia2 and Wen Yang2
1Key Laboratory of Computer Vision and System, Ministry of Education, Tianjin University of Technology, 300384, Tianjin, China
2Tianjin Key Laboratory of Intelligence Computing and Novel Software Technology, Tianjin University of Technology, 300384, China

*Corresponding author e-mail: daimin@tjut.edu.cn

Abstract. Spike sorting is difficult when there is high waveforms similarity between different spike or when there is a large number of superimposed spikes in the sample. A new sample optimization method is proposed in the paper, called window-gradient feature. Every spike waveform is segmented into successive fragments in terms of the width σ, and each segment is called a window. Then calculate the gradient change of each window and use the ratio as the new alternative feature of the window. Finally, all window-gradient features of the spike waveforms are used to replace the original waveform features for spike sorting. The method is verified on the simulation data at different SNR. The experimental results show that when using SVM for spike sorting, the optimization effect of the proposed method is better than PCA, especially for data sets data sets with high noise or large sampling waveform similarity.

1. Introduction

In the neural system, electrical impulses produced by neurons are called spikes. It is an important way to reveal human perception and cognitive activities to study the spike’s activity regulation. Extracellular recording can be used to record such activity of several neurons near an electrode. To understand the nervous system better, researchers need to know which spike is generated by which neuron. Therefore, we need techniques to detect the spikes generated by the same neuron from the original recording signals, collectively known as “spike sorting”.

Generally, the amplitude and the shape of spike produced by same neuron is similar, and different neurons produce different spike waveforms, which is the basis for spike sorting [1]. One of the commonly used approaches for spike sorting is the clustering of spikes [2,3]. There are many methods for extracting spike features, such as principal component analysis [3,4], wavelet transformation analysis[5], independent component analysis[6], etc. These optimization methods combined with classification algorithms can not only reduce the dimension of data, but also improve the performance of classification. However, when spike waveforms produced by different neurons are very similar or there are many superimposed spikes, the performance of the above method will be seriously degraded [7]. In recent years, sparse coding [8], deep learning[9] have also been applied to spike sorting, and has shown better performance. However, with the increase of the number of neurons, their complexity would not be overlooked.

A window-gradient feature representation is proposed in the paper. Each spike waveform is segmented into continuous segments at specified intervals σ, each of which is called a window. The gradient change of each window is calculated as a new feature for spike sorting. The method could
highlight the gradient change of a part of the spike waveform and reduce the noise interference, so it can improve the accuracy of spike sorting.

2. Methods

2.1. Waveform Features of Spike

A spike is a short and sharp electrical signal fired by the neurons in response to a stimulus. Spikes coming from different cell types are usually not alike concerning their shape. In many cases, however, the spikes produced by two different neurons are similar in shape and size (see Figure 1). This happens when the neurons are similar in shape and about same distant from the recording electrode [4].

![Figure 1. Two spikes with similar waveform](image)

The most obvious feature of a spike waveform is its positive peak and negative valley amplitude. We can see from a large number of experiments and observations that, even if the waveforms of different spikes are similar, the gradient changes at peaks and valleys may be different.

As shown in Figure 1, the shape and size of two spikes are similar, but the gradient change at the inflection (such as A, B and C points) are different. Some varies relatively smoothly, and some varies relatively sharply. If we can highlight the difference, this will help to improve the classification accuracy of spike sorting.

As the data acquisition equipment, Multi-electrode array (MEA) has been widely used for neurological study. If there are many active (spiking) neurons near one microelectrode pin, the electrode will record the overlapped signal of many neurons. The overlapped signals make it difficult for spike sorting, because the overlapped spikes make the part of the single spike information loss. Ordinary spike sorting methods are difficult to separate overlapped waveforms.

2.2. Window-Gradient Features

For different spikes waveform, the peaks and valleys gradient change are different, even if the width and height of them are similar. According to the characteristic, we use gradients in a series of windows to represent the change rapid of a spike waveform. Each spike waveform is segmented into successive segments at the specified interval $\sigma$, called a window. The gradient change of each window is calculated as the new features for spike sporting. As shown in Figure 2, if A is the maximum amplitude in the window and B is the minimum, the gradient of current window is the ratio of the the difference between A and B and the window’s width.

![Figure 2. Window Gradient Feature](image)
Let \( x_i(n) = (x(0), x(1), \ldots, x(n-1)) \) represents the \( i \)-th original spike sample, and \( n \) is the dimension of \( x \). Let \( y_i(j) = (y(0), y(1), \ldots, y(k-1)) \) represents the sample after extracting the window-gradient (WG) feature from \( x_i \), and \( k \) is the dimension of \( y \). Then the \( j \)-th component of \( y_i(j) \) can be calculated by formula (1).

\[
y_i(j) = \frac{\max \{x_i(j(\sigma-1)), \ldots, x_i((j+1)(\sigma-1))\} - \min \{x_i(j(\sigma-1)), \ldots, x_i((j+1)(\sigma-1))\}}{\sigma}, \quad 0 \leq j \leq \frac{n+1}{\sigma} \tag{1}
\]

where \( \sigma \) is window’s width, which is usually set as a fixed time interval or sampling interval. \( \sigma \) can not be too big, otherwise, you can easily lose the details of the waveform. Our experiments on a large number of different data show that the best result appears usually when \( \sigma \) is set to 2~4.

It can be seen from formula (1), that when the window width \( \sigma \) is fixed, \( y_i(j) \) will increase with the increase of the amplitude gradient in the window, reflecting the sharp change of the waveform.

Figure 3 shows the window-gradient (WG) features. Two raw spike waveforms are similar, but the differences in shape are enlarged obviously by using WG features. This will help to improve the accuracy of classification.

![Figure 3](image1.jpg)

**Figure 3.** Window-gradient feature for similar spike (\( \sigma=2 \))

![Figure 4](image2.jpg)

**Figure 4.** Window-gradient feature for overlapped spikes (\( \sigma=2 \))
For overlapped spikes, part of the lost information caused by superposition could be supplied by window-gradient feature representation method. In figure 4, spike A and B are different spikes, and (c) is the overlapped spike waveform of A and B. We can see from figure 4(f) ($\sigma=2$), the window-gradient feature complements part of the lost details of the overlapped spikes, and it will be more benefit for overlapped spike sorting.

3. Experiments and Results

Our experimental data are simulation data sets provided by Wave-clus software[10]. The simulation data sets include four sets, and each set contains simulated continuous neural signals with different signal-noise-ratio (SNR). Each spike in the simulated data is marked by which neuron is producing it. This makes it easy to compare experimental results. In our experiments, there are three type spikes in each data set, which are labeled with S1, S2 and S3 respectively. For each spike, the spike type label and the information specified whether a spike is overlapped are stored. If the spike is an overlapped spike, the time interval of the overlapped is also given. The number of spike waveforms in each is between approximately 3200 and 3500. We extract a spike waveform according to the spike fire time specified by spike_times. We take 19 points before the fire time point and 44 points after the fire time point, which forms a 64 points spike sample.

The following two experiments will give an explanation about the spike optimization result of this method and the classification result combine with the SVM.

3.1. Optimization results of window-gradient features representation

In order to test the optimization result of this method, we used Hausdorff distance to measure the similarity between different spikes in our experiment. The greater of the Hausdorff distance, the less similar of the two different spikes, and the better of the optimization [11].

We verified the proposed method on each dataset described above, and compared it with the principal component analysis (PCA) method. Some experimental results are shown in Table 1 ($\sigma=2$).

| Data                  | Method | S1 and S2 | S1 and S3 | S2 and S3 |
|-----------------------|--------|-----------|-----------|-----------|
| C_Easy2_noise02       | None   | 0.0475    | 0.0462    | 0.0471    |
|                       | PCA    | 0.0591    | 0.1205    | 0.1626    |
|                       | WG     | 0.1998    | 0.2       | 0.157     |
| C_Difficult2_noise005 | None   | 0.0889    | 0.0862    | 0.0467    |
|                       | PCA    | 0.0841    | 0.1162    | 0.0639    |
|                       | WG     | 0.0879    | 0.1413    | 0.2291    |
| C_Difficult2_noise01  | None   | 0.0875    | 0.0864    | 0.0447    |
|                       | PCA    | 0.0595    | 0.1636    | 0.1426    |
|                       | WG     | 0.2162    | 0.2258    | 0.1655    |

In table 1, S1, S2 and S3 represent the three type spikes in each dataset. The distance in the table is the Hausdorff distance between various type template samples. The template samples of each type are got by average and normalization method. The line “none” is the distance between original spike samples without optimized representation. The distance in other two lines is the distance after using the PCA and WG features optimization method respectively. In the experiment, the window width $\sigma$ is set to 2.

We can see from table 1, PCA and WG features optimization method can increase the distance between different types of template, so it will be benefit for spike sorting. Comparing with PCA method, using WG features can highlight the peaks and valleys shape change, so the Hausdorff distance between different spike types increased more obvious.
3.2. Spike sorting using SVM

As a statistical learning method proposed by Cortes and Vapnik in 1995, support vector machine (SVM) has advantages in small sample, non-linear and high-dimensional pattern recognition [12]. We look for the support vector by fixing the maximum interval between different types of samples and decide the optimal separating surface, use this separating surface to classify the test sample, the number of the support vector will affect the final classification speed.

In the experiment, SVM, PCA combined with SVM and WG combined with SVM are used to classify spike samples, and the number of support vector classification, the classification accuracy and running time of the results are compared.

This experiment use six sets of data with different similarities and different noise levels, and each data contains three types of spike with large number of overlapped samples. The top 500 samples are used as the training samples and the rest as the test samples. width \( \sigma \) is set 2 in our experiments.

Experimental results are shown in Table 2.

| Data                    | Method       | Support vector | Accuracy (%) | Running time (s) |
|-------------------------|--------------|----------------|--------------|------------------|
| C_Easy2_noise005        | SVM          | 91             | 94.33%       | 0.164            |
|                         | PCA+SVM      | 72             | 96.32%       | 0.209            |
|                         | WG+SVM       | 58             | 98.25%       | 0.238            |
|                         | SVM          | 107            | 98.91%       | 0.185            |
|                         | PCA+SVM      | 87             | 98.48%       | 0.226            |
|                         | WG+SVM       | 71             | 99.54%       | 0.291            |
| C_Easy2_noise01         | SVM          | 125            | 97.17%       | 0.243            |
|                         | PCA+SVM      | 101            | 97.98%       | 0.368            |
|                         | WG+SVM       | 83             | 98.07%       | 0.383            |
| C_Easy1_noise015        | SVM          | 84             | 98.36%       | 0.145            |
|                         | PCA+SVM      | 62             | 98.6%        | 0.199            |
|                         | WG+SVM       | 49             | 99.48%       | 0.385            |
| C_Difficult2_noise005   | SVM          | 267            | 98.23%       | 0.671            |
|                         | PCA+SVM      | 183            | 98.34%       | 0.713            |
|                         | WG+SVM       | 94             | 99.08%       | 0.678            |
| C_Difficult2_noise015   | SVM          | 330            | 67.12%       | 0.732            |
|                         | PCA+SVM      | 206            | 76.8%        | 0.768            |
|                         | WG+SVM       | 135            | 94.10%       | 0.689            |

We can see from the Table 2, compare to only using SVM and using PCA to optimization, this method proposed in this paper increase the running time. But the accuracy increase obviously, especially for the data set which the noise level and the similarity of the sample is larger and the traditional method is difficult to classify, this method is obviously benefit to improve the classification accuracy. Otherwise reducing the number of support vector effectively reduce the complexity of computing and accelerate the speed of classification of SVM.

4. Summary and Conclusion

The spike sorting is the premise of studying the information processing mechanism of neurons. If two spikes are different in waveform, it is possible to make the correct classification using the aforementioned methods, but the performance of the above method will be seriously degraded when
two spikes shapes is very similar or fire simultaneously. Through analyzing the characteristics of spikes waveform, a new optimization method for feature representation is proposed in the paper. Each spike waveform is segmented into continuous segments of width $\sigma$. The gradient changes of each segment are calculated as a new feature for spike sporting. The method is tested on simulation data provided by Wave_clus. Experiment results show that the method could highlight the gradient change degree of a spike waveform and reduce the noise interference, so it is obviously benefit to improve the accuracy of spike sorting. When using SVM for spike sorting, the optimization effect of the proposed method is better than PCA, especially for data sets data sets with high noise or large sampling waveform similarity.

Acknowledgments
This work was financially supported by Tianjin Natural Science Foundation under Grant No. 16JCYBJC15300.

References
[1] YuanYuan, ChenhuiYang, Jennie Si. The M-Sorter: an automatic and robust spike detection and classification system[J]. Journal of Neuroscience Methods. 2012, 210(2): 281-290.
[2] James Zhang, Thanh Nguyen, Steve Cogill, et al. A review on cluster estimation methods and their application to neural spike data [J]. Journal of Neural Engineering. 2018 (15):1-18.
[3] O Gafurov, A Zakharov, K Koroleva, R Giniatullin. Improvement of Nociceptive Spike Clusterization with Shape Approximation [J], BioNanoScience, 2017(7) :565–569.
[4] Jianwei Li, Youhua Wang, Guilong Zong. Support vector machine method using in EEG signals study of epileptic spike[C]. 2009 International Conference on Machine Learning and Cybernetics, 2009(2):1241-1245.
[5] Junwei Duan, Long Chen, C. L. Philip Chen. A new method based on wavelet and greedy pursuit analysis for neuro-spike detection[C]. Proceedings 2014 International Conference on Informative and Cybernetics for Computational Social Systems.2014:74-78.
[6] Marcus Vinicius Lopes, Enio Aguiar, etc. ICA feature extraction for spike sorting of single-channel records[C]. 2013 Biosignals and Biorobotics Conference: Biosignals and Robotics for Better and Safer Living (BRC), 2013:1-5.
[7] Yun-Yu Chen, Tung-Chien Chen, and Liang-Gee Chen. Accuracy and power tradeoff in spike sorting microsystems with cubic spline interpolation[C]. Proceedings of 2010 IEEE International Symposium on Circuits and Systems, 2010, 1508-1511.
[8] Kai Yang, Haifeng Wu, Yu Zeng. Sparse coding approaches for neuron spike sorting[C]. 2016 IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC), 2016, 600-604.
[9] Kai Yang, Haifeng Wu, Yu Zeng. A Simple Deep Learning Method for Neuronal Spike Sorting[J]. Journal of Physics. 2017(910): 012062
[10] Dai, Min, and Jing Luo. A Robust Method for Spike Sorting with Overlap Decomposition [J]. Journal of Computers, 2014. 9(3):733-740.
[11] Xinshu Jiang, Yue Lv. Improved Approach to Image Matching Based on Weighted Hausdorff Distance [J]. Application Research of Computers, 2007, 24(4):182-183,238
[12] Kyung Hwan Kim, Sung Shin Kim, Sung June Kim. Advantage of support vector machine for neural spike train decoding under spikesorting errors[C]. IEEE Engineering in Medicine and Biology 27th Annual Conference, 2005: 5280-5283.