

SpikeGUI: Software for Rapid Interictal Discharge Annotation via Template Matching and Online Machine Learning

Jing Jin¹, Justin Dauwels², Sydney Cash³, and M. Brandon Westover⁴

Abstract—Detection of interictal discharges is a key element of interpreting EEGs during the diagnosis and management of epilepsy. Because interpretation of clinical EEG data is time-intensive and reliant on experts who are in short supply, there is a great need for automated spike detectors. However, attempts to develop general-purpose spike detectors have so far been severely limited by a lack of expert-annotated data. Huge databases of interictal discharges are therefore in great demand for the development of general-purpose detectors. Detailed manual annotation of interictal discharges is time consuming, which severely limits the willingness of experts to participate. To address such problems, a graphical user interface “SpikeGUI” was developed in our work for the purposes of EEG viewing and rapid interictal discharge annotation.

“SpikeGUI” substantially speeds up the task of annotating interictal discharges using a custom-built algorithm based on a combination of template matching and online machine learning techniques. While the algorithm is currently tailored to annotation of interictal epileptiform discharges, it can easily be generalized to other waveforms and signal types.

I. INTRODUCTION

Interictal discharges [1] are essential in the diagnosis and management of epilepsy. However, they are difficult to detect in a consistent manner. Attempts have been made to create automatic systems and algorithms [2], [3], [4], which are not fully tested nor accepted universally. The biggest hurdle to achieving a strong algorithm for detection is the lack of a sufficient database of annotated EEG records.

There are many ways we could go about establishing a foundation for this problem. We could have a number of experts manually create a database. However, detailed manual annotation of interictal discharges is slow and boring, especially for records with many interictal discharges (several thousands per hour), which severely limits the willingness of experts to participate. Alternatively, we could employ an existing detection system supplied commercially to create a database. But the sensitivity and specificity of these systems are poorly documented. As a result, we would not be able to use this as anything approaching a gold standard.

To this end, we have come up with a hybrid approach to reduce the labor and speed the process of acquiring expert

annotations of EEG data. It is a MATLAB-based graphical user interface, named “SpikeGUI”, which is designed for EEG viewing, and rapid interictal discharge annotation. It is based on the observation that, within patients, interictal discharges tend to be fairly stereotyped, which suggests that selecting one example as a template can enable instantly and automatically extracting many more candidate matches, which can then be rapidly accepted or rejected by an expert. The rapid vote/feedback in turn suggests that annotation can be further speeded up by being cast into an online learning task, which provides progressively higher recommendations.

“SpikeGUI” is a full-featured EEG viewer that is designed to be easy to use and allow for high speed viewing. “SpikeGUI” employs a custom-built signal processing algorithm consisting of template matching [5] and online machine learning [6] to ensure rapid interictal discharge annotation.

This paper is organized as follows. In section II, we briefly discuss our scalp EEG data and techniques involved in “SpikeGUI”. In section III, validation and annotation results are presented, and in section IV concluding remarks and recommendations for future work are offered.

II. MATERIALS AND METHODS

A. Epileptic Scalp EEG

We consider here data from 303 patients with known epilepsy who underwent scalp EEG recording at MGH with international 10-20 system of electrode placement. In each case, a 30-min EEG record with 19 scalp electrodes was used. EEG recordings were down-sampled to 128Hz, and band-pass filtered between 0.1 and 64Hz. A notch filter was applied to remove the 60Hz power-line interference.

B. Rapid Interictal Discharge Detection

There are 2 major techniques involved in rapid interictal discharge detection: template matching and online machine learning. Template matching is applied to generate a list of interictal discharge candidates based on the z-normalized Euclidean distance computed with respect to a given interictal discharge template. Online machine learning is used afterwards to refine the ranking in the list for further selection.

1) *Template Matching*: Template matching (TM) [7], [8] is carried out based on the z-normalized Euclidean distance. Euclidean distance between 2 samples $p, q \in \mathbb{R}^{1 \times n}$ denoted

¹J.Jing is with School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore jing0006@e.ntu.edu.sg

²J.Dauwels is with School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore justin@dauwels.com

³S.Cash is with the Neurology Department, Massachusetts General Hospital, Boston, MA, USA, and Harvard Medical School, Cambridge, MA, USA Scash@partners.org

⁴M.B.Westover is with the Neurology Department, Massachusetts General Hospital, Boston, MA, USA, and Harvard Medical School, Cambridge, MA, USA mwestover@partners.org

by $\|p - q\|$ is computed as:

$$\|p - q\|^2 = \sum_{i=1}^n (p_i - q_i)^2. \quad (1)$$

It is commonly used to measure the similarity between samples. For each record, a distance look-up table (LUT) is computed beforehand with respect to the same reference randomly selected. To reduce computational complexity, the triangle inequality [8], [9] is applied to reject samples far away from the given template as shown in Fig. 1, and narrow down the range of search to a small group of samples. The accepted samples are further ranked according to the Euclidean distance to the given template in ascending order.

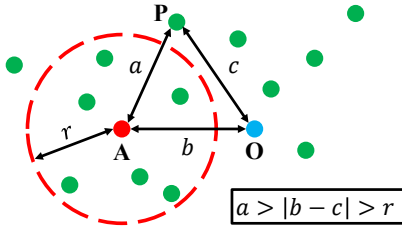


Fig. 1: The triangle inequality to reject samples outside the accepted region ($a > r$); with O denoting the reference for LUT computation, P the sample, A the template, (a, b, c) the sides of $\triangle OPA$, and r the radius of accepted region.

2) **Online Machine Learning:** Online machine learning (OML) is a model of induction that learns sequentially [10]. OML is applied after TM to further refine the ranking in the list of candidate waveforms. The key defining characteristic of OML is that the true label of the instance is revealed soon after the prediction is made, to refine the prediction hypothesis for future trials. Due to continual label feedback, the online learning algorithms are able to adapt and learn in difficult situations.

The goal of the algorithm is to minimize some performance criteria which are algorithm specific. In this paper, the MATLAB-based toolbox LIBOL [11] is applied to provide a collection of various OML algorithms. Specifically, the second formulation of “Soft Confidence-weighted learning” (SCW-II) [10] is selected as our OML algorithm with the best performance, which will be explained in detail in Section III.

Mathematically, at time step t , SCW-II receives the incoming sample x_t , and predicts its label \hat{y}_t . The true label y_t is then revealed and the loss $l(y_t, \hat{y}_t)$ is determined. Assuming a Gaussian distribution of weights with mean μ and covariance Σ , the loss function of SCW is defined as:

$$l^\phi(\mathcal{N}(\mu, \Sigma); (x_t, y_t)) = \max(0, \phi\sqrt{x_t^\top \Sigma x_t} - y_t \mu x_t), \quad (2)$$

with ϕ denoting the inverse of the cumulative function of the normal distribution. The optimization problem can be written as:

$$\begin{aligned} (\mu_{t+1}, \Sigma_{t+1}) = \arg \min_{\mu, \Sigma} & D_{KL}(\mathcal{N}(\mu, \Sigma) \parallel \mathcal{N}(\mu_t, \Sigma_t)) \\ & + Cl^\phi(\mathcal{N}(\mu, \Sigma); (x_t, y_t))^2, \end{aligned} \quad (3)$$

with D_{KL} denoting the Kullback-Leibler divergence [12], C the parameter to tradeoff the passiveness and aggressiveness. The closed-form solution of the optimization problem in Eqn. (3) is:

$$\mu_{t+1} = \mu_t + \alpha_t y_t \Sigma_t x_t \quad (4)$$

$$\Sigma_{t+1} = \Sigma_t - \beta_t \Sigma_t x_t^\top x_t \Sigma_t,$$

with α_t, β_t denoting the updating coefficients. The detailed proofs can be found in [10].

In our work, multiple time and frequency features relevant to interictal discharges are used for training OML: (i) peak and (ii) peak-to-trough values, (iii) steepness (defined as the time taken to drop from the maxima to its 25%), (iv) variance, and (v) power in frequency band between 20 to 80Hz. Features (i)-(iv) are extracted from the smoothed nonlinear energy operator (sNLEO) [13] of the waveform, while feature (v) is obtained directly from the waveform itself.

C. The “SpikeGUI” System

The “SpikeGUI” graphical user interface (GUI) consists of two sub-GUIs: the navigation GUI (Fig.2) for EEG viewing and annotation; and the minor GUI (Fig.3) to display the list of candidate waveforms located by TM+OML.

After importing the EEG recording, it is shown in the navigation window along with previous annotations if any. Basic navigation functions are available such as shifting along time either at different step size (5s or 10s) or via a swift slider, amplitude scaling up/down, montage swap (monopolar, common average, and bipolar), and manual annotation. The button “Auto-Template Match” is meant for the core algorithm for rapid annotation, i.e., TM+OML. To execute this function, one has to manually select an interictal discharge template by left clicking the mouse at the interictal discharge (right clicking to un-select) before pressing the button. A list of SpikeGUI-recommended waveforms with respect to the template will pop up immediately for further selection (See Fig. 3).

The waveforms are ranked according to the similarity to the template resulting from TM+OML in descending order. Interictal discharges newly selected will be annotated automatically in the navigation window as shown in Fig. 2. Apart from navigating along time with fixed time step-size or sliders, interictal discharges annotated can be reviewed by buttons “previous spike” and “next spike”, which jumps directly to the nearest (± 1) interictal discharge marker found in the record. Annotation status in terms of total current interictal discharge count and OML classification rate are shown at the top for the purpose of supervision.

“SpikeGUI” creates an individual record for each user, and allocates memory to export and store interictal discharge markers instantly. One can cease in the middle of annotation and come back to start from where one left off before. It also allows to load and view markers from others. With easy and quick addition and deletion of discharge markers, “SpikeGUI” becomes a handy tool for Ground

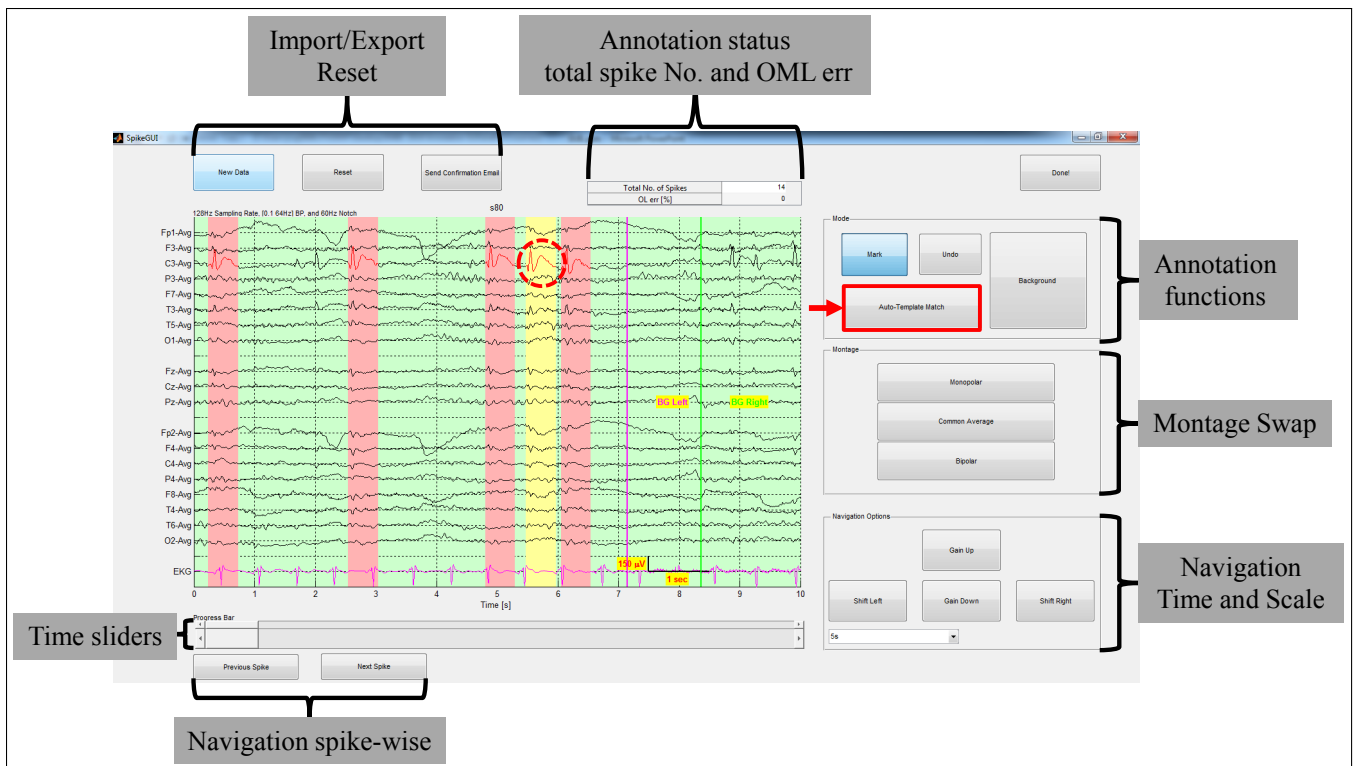


Fig. 2: A labeled screenshot of the navigation window of “SpikeGUI”. The EEG recordings are displayed with old markers if any: interictal discharges are labeled in red with pink background, and baselines are marked by pairs of magenta-green lines. Manual selection/annotation of interictal discharges/backgrounds can be easily done by left clicking the mouse at the target (right clicking to un-select). The current template manually selected is labeled in red with yellow background.

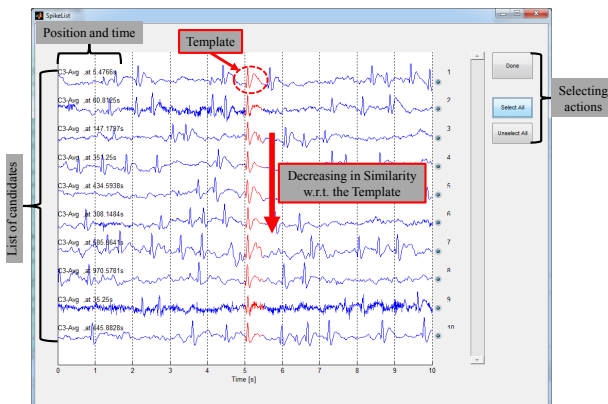


Fig. 3: List of candidate waveforms for further selection by checking radio buttons. The waveforms are ranked in a descending order of similarity to the template.

Truth Generation, i.e., the markers have to be agreed by 3 experts simultaneously, which is crucial for developing and validating detection algorithms. “SpikeGUI” is written in MATLAB® [14] and distributed using a run time compiler freely available from The Mathworks. “SpikeGUI” can be run on both Windows and Linux OS.

III. RESULTS

In this work, we developed an algorithm for efforts to annotate epileptiform discharges in clinical EEG recordings. It is based on the observation that, within patients, interictal

discharges tend to be fairly stereotyped, i.e., close in z -normalized Euclidean distance. It suggests that selecting one example as a template can extract many more matches rapidly (see Fig. 4).

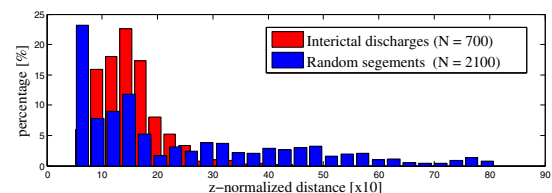


Fig. 4: Percentage histograms of z -normalized Euclidean distance of interictal discharges (red) and randomly selected segments (blue) extracted from the same record.

Although fast, TM has its own drawbacks, resulting in occasional bad ranking in the list of waveforms (see Fig. 5). Simple Euclidean distance may not be adequate to represent the similarity and reveal the important morphological patterns of the interictal discharges, especially the main peaks. It may lead to low acceptance rate and consequentially slow down the annotation process.

With feedback from the user, the annotation can be cast into an OML task. By continuously learning from previous annotations, the current ranking in the list can be refined by applying OML (see Fig. 5). To choose the proper OML algorithm, benchmark experiments were carried out with 14 different OML algorithms: Perceptron [15], RAMMA and agg-RAMMA [16], OGD [17], PA-

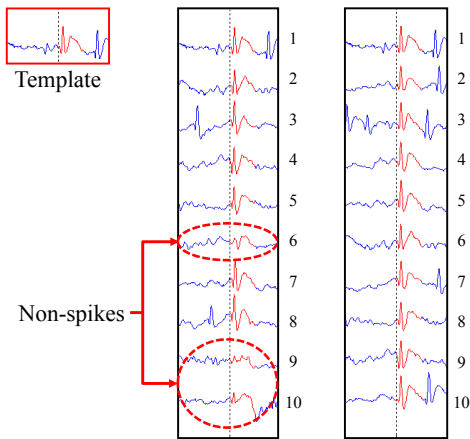


Fig. 5: OML improves the ranking: (left) suboptimal ranking by TM alone; (right) improved ranking due to OML.

I and PA-II [18], SOP [19], CW [20], IEL-LIP [21], NHERD [22], AROW [23], NAROW [24], SCW-I and SCW-II [10], using 100 interictal discharges and 100 non-ictal discharges from the same record. As shown in Fig. 6, “SCW II” outperformed the others with the lowest mistake rate, less no. of updates, and relatively low time cost.

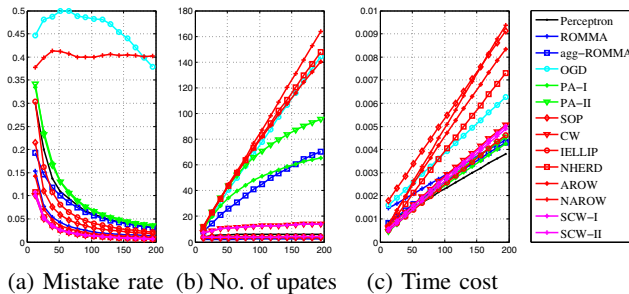


Fig. 6: Comparison of various OML algorithms; with performance criteria (a) cumulative rate of mistake, (b) cumulative no. of updates, and (c) cumulative time cost vs. the no. of samples.

To study the annotation speed of “SpikeGUI” in different scenarios, we had 3 experts annotate the same record (with 900⁺ interictal discharges) using “SpikeGUI” with manual annotation alone, TM alone, and TM with OML. The time costs are summarized in Tab. I.

	Manual	TM	TM+OML
Expert 1	180 min	80 min	40 min
Expert 2	150 min	85 min	50 min
Expert 3	200 min	90 min	55 min

TABLE I: Time costs of annotation experiments in different scenarios: manual annotation, TM alone, and TM with OML.

IV. CONCLUSIONS

In this work, we developed a MATLAB-based graphical user interface “SpikeGUI” for rapid interictal discharge annotation. “SpikeGUI” employs a custom-built signal processing algorithm toward automated EEG analysis, consisting of techniques such as template matching and online machine learning. While the algorithm is currently tailored to annotation of interictal epileptiform discharges, it can easily be generalized to other waveforms and signal types.

We have already extracted 35000⁺ interictal discharges from 303 patients, and the number continues to grow. With the database built at hand, we will push the project to develop a general-purpose interictal discharge detector in the near future.

REFERENCES

- [1] G. Chatrian, L. Bergamini, M. Dondey, D. Klass, M. Lennox-Buchthal, and I. Petersen, “A glossary of terms most commonly used by clinical electroencephalographers,” *Electroencephalogr Clin Neurophysiol*, vol. 37, no. 5, pp. 538–548, 1974.
- [2] S. B. Wilson and R. Emerson, “Spike detection: a review and comparison of algorithms,” *Clinical Neurophysiology*, vol. 113, no. 12, pp. 1873–1881, 2002.
- [3] M. Adjouadi, D. Sanchez, M. Cabrerizo, M. Ayala, P. Jayakar, I. Yaylali, and A. Barreto, “Interictal spike detection using the Walsh transform,” *Biomedical Engineering, IEEE Transactions on*, vol. 51, no. 5, pp. 868–872, 2004.
- [4] A. Ossadchi, S. Baillet, J. Mosher, D. Thyerlei, W. Sutherling, and R. Leahy, “Automated interictal spike detection and source localization in magnetoencephalography using independent components analysis and spatio-temporal clustering,” *Clinical Neurophysiology*, vol. 115, no. 3, pp. 508–522, 2004.
- [5] R. Brunelli and T. Poggio, “Face recognition: Features versus templates,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 15, no. 10, pp. 1042–1052, 1993.
- [6] M. G. Moore and G. Kearsley, *Distance education: A systems view of online learning*. Cengage Learning, 2011.
- [7] H. Breu, J. Gil, D. Kirkpatrick, and M. Werman, “Linear time Euclidean distance transform algorithms,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 17, no. 5, pp. 529–533, 1995.
- [8] A. Mueen, E. J. Keogh, Q. Zhu, S. Cash, and M. B. Westover, “Exact discovery of time series motifs,” in *SDM*, pp. 473–484, SIAM, 2009.
- [9] C. Elkan, “Using the triangle inequality to accelerate k-means,” in *ICML*, vol. 3, pp. 147–153, 2003.
- [10] J. Wang, P. Zhao, and S. C. Hoi, “Exact soft confidence-weighted learning,” *arXiv preprint arXiv:1206.4612*, 2012.
- [11] S. C. Hoi, J. Wang, and P. Zhao, “Libol: A library for online learning algorithms,” *Nanyang Technological University*, 2012.
- [12] D. Polani, “Kullback-leibler divergence,” *Encyclopedia of Systems Biology*, pp. 1087–1088, 2013.
- [13] S. Mukhopadhyay and G. Ray, “A new interpretation of nonlinear energy operator and its efficacy in spike detection,” *Biomedical Engineering, IEEE Transactions on*, vol. 45, no. 2, pp. 180–187, 1998.
- [14] I. MathWorks, *MATLAB: the language of technical computing. Desktop tools and development environment, version 7*, vol. 9. MathWorks, 2005.
- [15] F. Rosenblatt, “The perceptron: a probabilistic model for information storage and organization in the brain,” *Psychological review*, vol. 65, no. 6, p. 386, 1958.
- [16] Y. Li and P. M. Long, “The relaxed online maximum margin algorithm,” *Machine Learning*, vol. 46, no. 1-3, pp. 361–387, 2002.
- [17] M. Zinkevich, “Online convex programming and generalized infinitesimal gradient ascent,” 2003.
- [18] K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer, “Online passive-aggressive algorithms,” *The Journal of Machine Learning Research*, vol. 7, pp. 551–585, 2006.
- [19] N. Cesa-Bianchi, A. Conconi, and C. Gentile, “A second-order perceptron algorithm,” *SIAM Journal on Computing*, vol. 34, no. 3, pp. 640–668, 2005.
- [20] M. Dredze, K. Crammer, and F. Pereira, “Confidence-weighted linear classification,” in *Proceedings of the 25th international conference on Machine learning*, pp. 264–271, ACM, 2008.
- [21] L. Yang, R. Jin, and J. Ye, “Online learning by ellipsoid method,” in *Proceedings of the 26th Annual International Conference on Machine Learning*, pp. 1153–1160, ACM, 2009.
- [22] K. Crammer and D. D. Lee, “Learning via Gaussian herding,” in *NIPS*, pp. 451–459, 2010.
- [23] K. Crammer, A. Kulesza, M. Dredze, et al., “Adaptive regularization of weight vectors,” *Machine learning*, vol. 91, no. 2, pp. 155–187, 2013.
- [24] F. Orabona and K. Crammer, “New adaptive algorithms for online classification,” in *NIPS*, pp. 1840–1848, 2010.