Iterative Learning Cascaded Multiclass Kernel Based Support Vector Machine for Neural Spike Data Classification

Amir Zjajo, Rene van Leuken
Circuits and Systems Group
Delft University of Technology
Mekelweg 4, 2628 CD Delft, The Netherlands

Abstract—**In this paper, we develop an iterative learning framework based on multiclass kernel support vector machine (SVM) for adaptive classification of neural spikes. For efficient algorithm execution, we transform a multiclass problem with the Kesler's construction and extend iterative greedy optimization reduced set vectors approach with a cascaded method. Since obtained classification function is highly parallelizable, the problem is sub-divided and parallel units are instantiated for the processing of each sub-problem via energy-scalable kernels. After partition of the data into disjoint subsets, we optimize the data separately with multiple SVMs. We construct cascades of such (partial) approximations and use them to obtain the modified objective function, which offers high accuracy, has small kernel matrices and low computational complexity.**

I. Introduction

The high density of neurons in neurobiological tissue require a large number of recording electrodes to be implanted into relevant cortical regions for accurate representation of neural activity in freely moving subjects (e.g., for spatially broad analysis of neuronal synchronization), and to allow the location controllability of the recording sites [1]. Monitoring the activity of large number of neurons is a prerequisite for understanding the cortical structures and can lead to a better comprehension of severe brain disorders, such as Alzheimer's and Parkinson's diseases, epilepsy, autism and psychiatric disorders [2] or to reestablish sensory (e.g. vision, hearing) or motor (e.g. movement, speech) functions [3]. However, very frequently an electrode records the action potentials from multiple surrounding neurons (e.g. due to the background activity of other neurons, slight perturbations in electrode position or external electrical or mechanical interference, etc.), and the recorded waveform/spikes consist of the superimposed potentials fired from these neurons.

The ability to distinguish spikes from noise [4], and to distinguish spikes from different sources from the superimposed waveform, therefore depends on both the discrepancies between the noise-free spikes from each source and the signal-to-noise level (SNR) in the recording system. The time occurrences of the action potentials emitted by the neurons close to the electrode are detected, depending on the SNR, either by voltage thresholding with respect to an estimation of the noise amplitude in the signal or with a more advanced technique, such as continuous wavelet transform [5].

This research was supported in part by the European Union and the Dutch government as part of the CATRENE program under Heterogeneous INCEPTION project.

978-1-4799-6926-5/15/$31.00 ©2015 IEEE

After the waveform alignment, to simplify the classification process, a feature extraction step, such as principal component analysis (PCA) [6] or wavelet decomposition [7] characterizes detected spikes and represent each detected spike in a reduced dimensional space, i.e. for a spike consisting of n sample points, the feature extraction method produces m variables ($m<n$), where m is the number of features. Based on these features the spikes are classified into m-dimensional clusters by k-means [8], expectation maximization (EM) [9], template matching [10], Bayesian clustering [11] and artificial neural network (ANN) with each cluster corresponding to the spiking activity of a single neuron.

The support vector machine (SVM) has been introduced to bioinformatics and spike classification/sorting [12]-[14] because of its excellent generalization, sparse solution and concurrent utilization of quadratic programming, which provides global optimization. This absence of local minima is a substantial difference from the artificial neural network classifiers. Like ANN classifiers, applications of SVMs to any classification problem require the determination of several user-defined parameters, e.g. choice of an appropriate kernel and related parameters, determination of regularization parameter (i.e. C) and a appropriate optimization technique. Correspondingly, SVM applies the structure risk minimization instead of the empirical risk minimization and solves the problems of non-linear, dimensionality curse efficiently. However, the methods [12]-[14] could not identify multiclass neural spikes nor could they decompose overlapping neural spikes resulting from variable triggering of data collection (e.g. due to noise or other spike events leading to premature or delayed waveform). Recording multiple spikes on a specific electrode can also create complex sums of neuron waveforms [15].

In this paper, we develop a neural spike classification framework based on multiclass kernel SVM that is able to accurately identify overlapping neural spikes even for low SNR. For efficient algorithm execution, we transform the multiclass problem with the Kesler's construction and extend iterative greedy optimization reduced set vectors approach with a cascaded method. The implementation results show that a significant gain on throughput can be obtained, while classification rate of the proposed method consistently outperforms benchmarked methods over the entire range of SNRs tested.

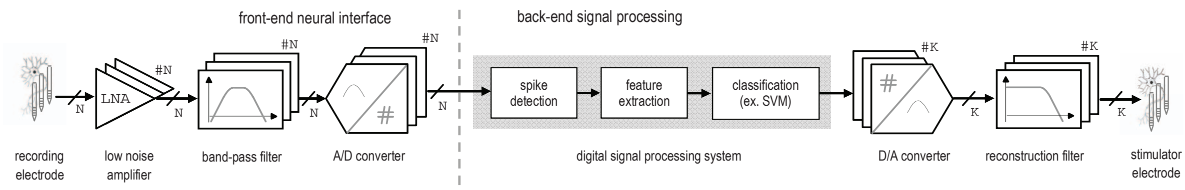

Figure 1: Block diagram of a brain-machine interface (BMI) with *N*-channel front-end neural recording interface and back-end signal processing.

II. Multiclass SVM Classification Framework

A. Architectural Overview of a Multichannel Neural Interface

With an increase in the range of applications and their functionalities, neuroprosthetic devices are evolving to a closed-loop control system [16] composed of a front-end neural recording interface and a back-end neural-signal processing. The brain-machine interface (BMI) architecture includes, additionally, a micro-stimulation module to apply stimulation signals to the brain neural tissues. The data acquired by the recording electrodes in brain-machine interface is conditioned and processed using analog circuits as illustrated in Figure 1. As a result of the small amplitude of neural signals and the high impedance of the electrode tissue interface, low-noise amplification (LNA) and band-pass filtering of the neural signals is performed before the signals are digitized by an analog to digital converter (A/D converter). The A/D converter output containing the time-multiplexed neural signals is fed to a back-end signal processing unit, which provides additional filtering and executes a spike detection. Threshold crossings of a local energy measurement [5] are used to detect spikes. A frequency-shaping filter significantly attenuates the low frequency noise and helps differentiating similar spikes from different neurons. The feature extraction based on maximum and minimum values of spike waveforms first derivatives [17] is employed due to its small computation and little memory requirement while preserving high information score. Neural spikes are classified with multi-class support vector machine. The relevant information is then transmitted to an outside receiver through the transmitter or used for *K*-channel stimulation in a closed-loop framework.

B. Multiclass Kernel SVM Training

The support vector machine is a linear classifier in the parameter space; nevertheless it becomes a nonlinear classifier as a result of the nonlinear mapping of the space of the input patterns into the high dimensional feature space. The classifier operations can be combined to realize variety of multi-class [18] and ensemble classifiers (e.g., classifier trees and adaptive boosting [19]). Instead of creating many binary classifiers to determine the class labels, we solve a multiclass problem directly [20] by modifying the binary class objective function and adding a constraining it for every class. The modified objective function allows simultaneous computation of multiclass classification [21].

Let us consider labelled training spike trains of N data points $\{y_k^{(i)}, x_k\}_{k=1,i=1}^{k=N,i=m}$, where x_k is the kth input pattern from n-dimensional space $\mathcal{R}^n$ and $y_k^{(i)}$ denotes the output of the ith output unit for pattern k, i.e. approach very similar to ANN methodology. The m outputs can encode $q=2^m$ different classes. The training procedure of the SVM corresponds to a convex optimization and amounts to solving a constrained quadratic optimization problem (QP); the solution found is, thus, guaranteed to be the unique global minimum of the objective function. To maximize the margin of $y(x)$, ω and b are chosen such that they minimize the nearest integer $||\omega||$ subject to the optimization problem formulated as [22]

$$\min_{\omega_i, b_i, \xi_{k,i}} J_{LS}^{(m)}(\omega_i, b_i, \xi_{k,i}) = \min \frac{1}{2}\sum_{i=1}^{m} \| \omega_i \|_2^2 + b_i^2 + C\sum_{k=1}^{N}\sum_{i=1}^{m}\xi_{k,i} \quad (1)$$

subject to the equality constraints

$$\begin{cases} y_k^{(1)}[\omega_1^T \varphi_1(x_k) + b_1] \geq 1 - \xi_{k,1}, \quad k = 1, \ldots, N \\ y_k^{(2)}[\omega_2^T \varphi_2(x_k) + b_2] \geq 1 - \xi_{k,2}, \quad k = 1, \ldots, N \\ \ldots \\ y_k^{(m)}[\omega_m^T \varphi_m(x_k) + b_m] \geq 1 - \xi_{k,m}, \quad k = 1, \ldots, N \end{cases} \quad (2)$$

where ω is a matrix of normal vectors (perpendicular to the hyperplane e.g. defined as $\omega x+b=0$), b is vector of biases, $C>0$ is the regularization constant, ξ is a vector of slack variables used to relax the inequalities for the case of non-separable data. The sum $\Sigma_{i,k}\xi_{k,i}$ is the cost function of spike trains whose distance to the hyperplane is less than margin $1/\|\omega\|$. In [23] is demonstrated that (1) is an acceptable formulation in terms of generalization errors though an additional term $b^2/2$ added to the objective. To solve the optimization problem we use the Karush-Kuhn-Tucker theorem [24]. We add a dual set of variables, one for each constraint and obtain the Lagrangian of the optimization problem (1)

$$\mathcal{L}^{(m)}(\omega_i, b_i, \xi_{k,i}; \alpha_{k,i}) = J_{LS}^{(m)} - \sum_{k=1}^{N} \alpha_{k,i}\{y_k^{(i)}[\omega_i^T \varphi_i(x_k) + b_i] - 1 + \xi_{k,i}\} \quad (3)$$

which gives as conditions for optimality

$$\begin{cases} \dfrac{\partial \mathcal{L}_1}{\partial \omega_i} = 0 \rightarrow \omega_i = \sum_{k=1}^{N} \alpha_{k,i} y_k^{(i)} \varphi_i(x_k) \\ \dfrac{\partial \mathcal{L}_1}{\partial b_i} = 0 \rightarrow \sum_{k=1}^{N} \alpha_{k,i} y_k^{(i)} = 0 \\ \dfrac{\partial \mathcal{L}_1}{\partial \xi_{k,i}} = 0 \rightarrow \alpha_{k,i} = \gamma \xi_{k,i} \end{cases} \quad (4)$$

for k=1,…,N and i=1,…,m. The offset of the hyperplane from the origin is determined by the parameter $b/||\omega||$. The function $\varphi(.)$ is a nonlinear function, which maps the input space into a higher dimensional space. To avoid working with the high-dimensional map φ, we instead choose a kernel function ψ by defining the dot-product in Hilbert space

$$\varphi(x)^T \varphi(x_k) = \psi(x, x_k) \quad (5)$$

enabling us to treat nonlinear problems with principally linear techniques. Formally, ψ is a symmetric, positive semi-definite Mercer kernel; the only condition required is that the kernel ψ satisfy a general positivity constraint [24].

To allow for mislabeled examples a modified maximum margin technique is employed [25]. If there exists no hyperplane $\omega x+b=0$ that can divide different classes, the objective function is penalized with non-zero slack variables ξ_i. The modified maximum margin technique than finds a hyperplane that separates the training set with a minimal number of errors ε and the optimization becomes a trade-off between a large margin and a small error penalty ε. The maximum margin hyperplane and consequently the classification task is than only a function of the support vectors

$$\max_{\alpha_k} Q_1(\alpha_k; \psi(x_k, x_l) = \sum_{k=1}^{N} \alpha_k - 1/2 \sum_{k,l=1}^{N} y_k y_l \psi(x_k, x_l) \alpha_k \alpha_l \quad (6)$$

$$\text{s.t.} \quad \alpha \in R^m \mid 0 \le \alpha_k \le C,\ k = 1, \dots, N,\ \sum_{k=1}^{N} \alpha_k y_k = 0$$

where α_k are weight vectors. The QP optimization task in (6) is solved efficiently using sequential minimal optimization, i.e. by constructing the optimal separating hyperplane for the full data set [26]. Typically, many α_k go to zero during optimization, and the remaining x_k corresponding to those $\alpha_k>0$ are called support vectors. To simplify notation, we assume that all non-support vectors have been removed, so that N_x is now the number of support vectors, and $\alpha_k>0$ for all k. The resulting classification function $f(x)$ in (6) has the following expansion

$$f(x) = \text{sgn} \sum_{k=1}^{N} \alpha_k y_k \psi(x, x_k) + b \quad (7)$$

where the support vector machine classifier uses the sign of $f(x)$ to assign a class label y to the object x [27].

The complexity of the computation of (7) scales with the number of support vectors. To simplify the kernel classifier trained by the SVM, we approximate an input pattern $x_k \in \mathcal{R}$ (using (7)), e.g. $\Psi=\sum\alpha_k\Phi(x_k)$ by a reduced set vectors $z_i \in \mathcal{R}$ e.g. $\Psi'=\sum\beta_k\Phi(z_k)$, $\beta_k \in \mathcal{R}$, where the weight vector $\beta_k \in \mathcal{R}$ and the vectors z_i determine the reduced kernel expansion. The problem of finding the reduced kernel expansion can be stated as the optimization task

$$\min_{\beta,z} \| \Psi - \Psi' \|^2 = \min_{\beta,z} \sum_{k,l=1}^{N_x} \alpha_k \alpha_l \psi(x_k \cdot x_l) + \sum_{k,l=1}^{N_z} \beta_k \beta_l \psi(z_k \cdot z_l) - 2 \sum_{k=1}^{N_x} \sum_{l=1}^{N_z} \alpha_k \beta_l \psi(x_k \cdot z_l) \quad (8)$$

Although Φ is not given explicitly, (8) can be computed (and minimized) in terms of the kernel and carried out over both the z_k and β_k. The reduced set vectors z_k and the coefficients $\beta_{l,k}$ for a classifier $f_l(x)$ are solved by iterative greedy optimization [28]

$$f_l(x) = \text{sgn} \sum_{k=1}^{m} \beta_{l,k} \psi(x, z_l) + b, \qquad l = 1, \dots, N_z \quad (9)$$

For a given complexity (i.e. number of reduced set vectors) the classifier provides the optimal greedy approximation of the full SVM decision boundary; the first one is the one which, using the objective function (8) is closest to the full SVM (7) constrained to using only one reduced set vector.

C. *Iterative Learning Cascaded Multiclass Kernel SVM Classification*

The transformation from the multiclass SVM problem in (1) to the single class problem is based on the Kesler's construction [25],[27]. Resulting SVM classifier is composed of the set of discriminant functions, which are computed as

$$f_l(x) = \sum_k \psi(x_k \cdot x) \sum_m \beta_k^m (\delta(l, y_k) - \delta(l, m)) + b_l \quad (10)$$

where the vector b_j, $m \in K$ is given by

$$b_l = \sum_k \sum_m \beta_k^m (\delta(l, y_k) - \delta(l, m)) \quad (11)$$

Since the data x_k appears only in the form of dot products in the dual form, we can construct the dot product (x_k, z_l) using the Kronecker delta, i.e., $\delta(k, l)=1$ for $k=l$, and $\delta(k, l)=0$ for $k \neq l$ and map it to a reproducing kernel Hilbert space such that the dot product obtains the same value as the function ψ. This property allows us to configure the SVM classifier via various energy-scalable kernels [29] for finding non-linear classifiers. For $\psi(.,.)$ one typically has the following choices: $\psi(x,x_k)=x_k^T x$ (linear SVM); $\psi(x,x_k)=(x_k^T x+1)^d$ (polynomial SVM of degree d); $\psi(x,x_k)=\tanh[\chi x_k^T x-\upsilon]$ sigmoid SVM); $\psi(x,x_k)=\exp\{-\gamma||x-x_k||^2\}$ (radial basis function (RBF) SVM); $\psi(x,x_k)=\exp\{-||x-x_k||/(2\sigma^2)\}$ exponential radial basis function (ERBF) SVM, and $\psi(x,x_k)=\exp\{-||x-x_k||^2/(2\sigma^2)\}$ Gaussian RBF SVM, where χ, υ, γ and σ are positive real constants. The performance of different kernel functions strongly depends on the characteristics of the application data. Consequently, the choice of kernel functions substantially affects the computational efficiency and memory requirements [30].

The reduced set vectors approach allows us to reduce the SVM complexity while maintaining accuracy. To increase the performance even further, we extend iterative greedy optimization reduced set vectors approach [28].

Accordingly, the reduced expansion is not evaluated at once, but rather in a cascaded way, such that in most cases a very small number of support vectors are applied. The computation of classification function $f_l(x)$ involves matrix-vector operations, which are highly parallelizable. Therefore, the problem is segmented into smaller ones and parallel units are instantiated for the processing of each sub-problem. Consider a set of reduced set vectors classification functions where the l-th function is an approximation with l vectors, chained into a sequence. After partition of the data into disjoint subsets, we iteratively train the SVM on subsets of the original data set and combine support vectors of resulting models to create new training sets [31]-[32]. A query vector is then evaluated by every function in the cascade and if classified negative the evaluation stops

$$f_{c,l}(x) = \mathrm{sgn}(f_1(x))\,\mathrm{sgn}(f_2(x))\ldots, \tag{12}$$

where $f_{c,l}(x)$ is the cascade evaluation function of (10). In other words, we bias each cascade level in a way that one of the binary decisions is very confident, while the other is uncertain and propagates the data point to the next, more complex cascade level. Biasing of the functions f is done by setting the parameter b to achieve a desired accuracy of the function on an evaluation set.

When a run through the cascade is completed, we combine the remaining support vectors of the final model with each subset from the first step of the first run. Frequently a single pass through the cascade produces satisfactory accuracy, however, if the global optimum is to be reached, the result of the last level is fed back into the first level to tests its fraction of the input vectors, i.e. whether any of the input vectors have to be incorporated into the optimization. If this is not valid for all input layer support vectors, the cascade is converged to the global optimum, else it proceeds with additional pass through the network.

III. Experimental Results

The test dataset is based on recordings from the human neocortex and basal ganglia. The dataset is constructed by combining three recordings, each containing the spiking activity of a different single unit. This was achieved by extracting the mean spike templates from the first two recordings and embedding these into the third recording at intervals corresponding to spike rates of 18 Hz and 20 Hz, respectively. In this manner the test dataset has three distinct spike clusters that are known but also contains realistic background activity. Simulated neural data was input to RTL simulations to obtain switching activity estimates for the design. These estimates were then annotated into the synthesis flow to obtain energy estimates for the digital spike-classification module.

Instead of thresholding the raw signal, we detect spikes in a more reliable way using threshold crossings of a local energy measurement of the bandpass filtered signal [5] (Figure 2 and Figure 3). The local energy threshold is equal to the squared average standard deviation of the signal defined by the noise properties of the recording channel and is equal to the minimal SNR required to be able to distinguish two neurons. Multiple single-unit spike trains are extracted from extracellular neural signals recorded from microelectrodes, and the information encoded in the spike trains is subsequently classified with RBF SVM kernel as illustrative example.

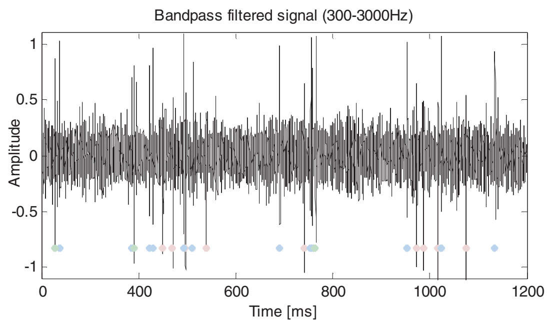

Figure 2: Spike detection from continuously acquired data, the y axis is arbitrary; threshold (line) crossings of a local energy measurement with a running window of 1ms.

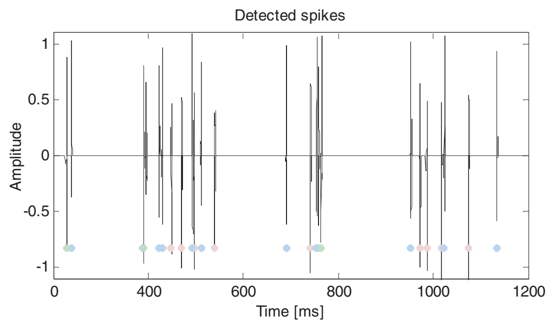

Figure 3: Spike detection from continuously acquired data, the y axis is arbitrary; detected spikes.

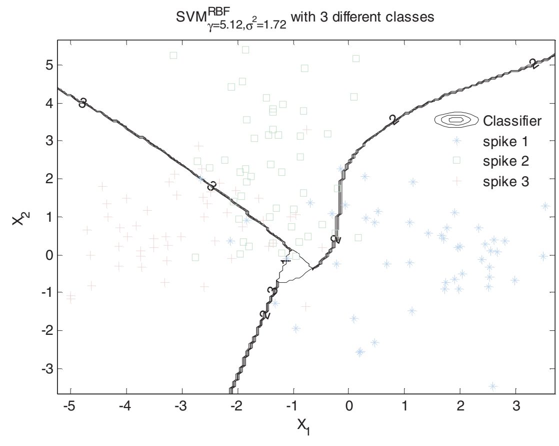

Figure 4: The SVM separation hypersurface for the RBF kernel.

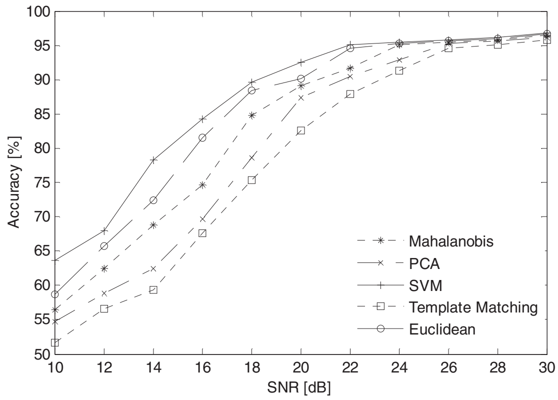

Figure 5: Effect of SNR on single spike sorting accuracy of the BMI system.

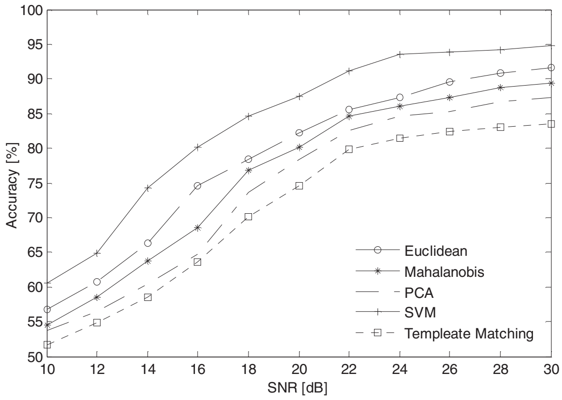

Figure 6: Effect of SNR on overlapping spikes of three classes on sorting accuracy of the BMI system.

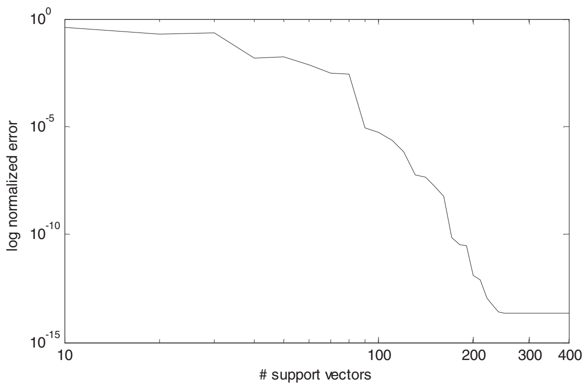

Figure 7: Log normalized error in reduced set model order reduction versus number of support vectors.

Each neuron action potential waveform is detected from a multiunit extracellular recording and assigned to one specific unit according to their waveform features. Since this procedure involve a substantial amount of error in the spike trains, and particularly when the background noise level is high, we measured testing classification error, training classification error, margin of the found hyperplane, and number of kernel evaluations.

Figure 4 gives a three classes classification graphical illustration, where the bold lines represent decision boundaries.

For a correctly classified example x_1, we have $\xi_1^{(1)}=0$ and $\xi_1^{(2)}=0$, i.e., no loss counted, since both $\varepsilon_{1,2}$ and $\varepsilon_{1,3}$ are negative. On the other hand, for an example x_2 that violates two margin bounds ($\varepsilon_{2,2},\varepsilon_{2,3}>0$), both methods generate a loss. The algorithm converges very fast at first steps and slows down as the optimal solution is approached. However, almost the same classification error rates were obtained for all the parameters $\varepsilon=[10^{-2},5\times10^{-3},10^{-3}]$, indicating that to find good classifier we do not need the extremely precise solution with $\varepsilon\rightarrow0$. The SVM performance is sensitive to hyper parameter settings, e.g., the settings of the complexity parameter C and the kernel parameter σ for the Gaussian kernel. As a consequence, hyper parameter tuning with grid search approach is performed before the final model fit. More sophisticated methods for hyper parameter tuning are available as well [33].

The SVM spike sorting performance has been summarized and benchmarked (Figure 5) versus four different, relatively computationally-efficient methods for spike sorting, e.g. template matching, principle component analysis, Mahalanobis and Euclidean distance. The performance is quantified using the effective accuracy, e.g. total spikes classified versus spikes correctly classified (excluding spike detection). The source of spike detection error is either the false inclusion of a noise segment as a spike waveform or the false omission of spike waveforms. These errors can be easily modeled by the addition or removal of spikes at random positions in time, so that the desired percentage of error ratio is obtained. In contrast, care should be taken in modeling spike classification errors, since an error in one unit may or may not cause an error in another unit. In all methods the suitable parameters are selected with which better classification performance are obtained. The SVM classifier consistently outperforms benchmarked methods over the entire range of SNRs tested, although it only exceeds the Euclidean distance metric by a slight margin reaching an asymptotic success rate of ~ 97%. The different SNRs in BMI have been obtained by superimposing attenuated spike waveforms such as to mimic the background activity observed at the electrode. If we increase the SNR of the entire front-end brain-machine interface, the spike sorting accuracy increases by up to 45% (depending on spike sorting method used). Similarly, the accuracy of the spike sorting algorithm increase with A/D converter resolution, although it saturates beyond 5-6 bit resolution, ultimately limited by the SNR. However, since the amplitude of the observed spike signals can vary, typically, by one order of magnitude, additional resolution is needed (i.e. 2-3 bit), if the amplification gain is fixed. Additionally, increasing the sampling rate of A/D converter improve spike sorting accuracy, since this captures finer features further differentiating the signals.

The sorting accuracy of the spike waveforms, which overlap at different sample points is illustrated in Figure 6. The correct classification rate of the proposed method is on average 4%-8% larger than that of other four methods.

If the training data contains the spike waveforms appearing in the process of complex spike bursts, we classify other distorted spikes generated by the bursting neurons firstly before resolving the problem of complex spike bursts partially. The performance of the four other methods is limited if the distribution of the background noise is non-Gaussian or if the multiple spike clusters are overlapped.

The number of support vectors required is partly governed by the complexity of the classification task. As the SNR decreases more support-vectors are needed in order to define a more complex decision boundary. For our dataset, the number of support-vectors required is within the range of 150-250, which is further reduced to 50-75 (for $\varepsilon=10^{-3}$) by an adapted reduced set model order reduction technique (Figure 7).

IV. Conclusion

The support vector machine has been introduced to bioinformatics and spike classification/sorting because of its excellent generalization, sparse solution and concurrent utilization of quadratic programming. In this paper, we develop a neural spike classification framework based on multiclass kernel SVM that is able to accurately identify overlapping neural spikes even for low SNR. As a result, the proposed method makes it feasible to successfully identify neural spikes, and avoid deteriorating power and chip area by overdesign.

References

[1] M.A. Lebedev, M.A.L. Nicolelis, "Brain-machine interfaces: Past, present and future," *Trends in Neurosciences*, vol. 29, no. 9, pp. 536-546, 2006.

[2] G. Buzsaki, "Large-scale recording of neuronal ensembles," *Nature Neuroscience*, vol. 7, pp. 446-451, 2004.

[3] F.A. Mussa-Ivaldi, L.E. Miller, "Brain-machine interfaces: Computational demands and clinical needs meet basic neuroscience," *Trends in Neuroscience*, vol. 26, no. 6, pp. 329-334, 2003.

[4] K.H. Lee, N. Verma, "A low-power processor with configurable embedded machine-learning accelerators for high-order and adaptive analysis of medical-sensor signals", *IEEE Journal of Solid-State Circuits*, vol. 48, no. 7, pp 1625-1637, 2013.

[5] K.H. Kim, S.J. Kim, "A wavelet-based method for action potential detection from extracellular neural signal recording with low signal-to-noise ratio," *IEEE Transactions on Biomedical Engineering*, vol. 50, pp. 999-1011, 2003.

[6] D.A. Adamos, E.K. Kosmidis, G. Theophilidis, "Performance evaluation of pca-based spike sorting algorithms," *Computer Methods and Programs in Biomedicine*, vol. 91, pp. 232-244, 2008.

[7] R.Q. Quiroga, Z. Nadasdy, Y. B. Shaul, "Unsupervised spike detection and sorting with wavelets and superparamagnetic clustering," *Neural Computation*, vol. 16, pp. 1661-1687, 2004.

[8] S. Takahashi, Y. Anzai, Y. Sakurai, "A new approach to spike sorting for multi-neuronal activities recorded with a tetrode-how ICA can be practical," *Neuroscience Research*, vol. 46, pp. 265-272, 2003.

[9] F. Wood, M. Fellows, J. Donoghue, M. Black, "Automatic spike sorting for neural decoding," *Proceedings of IEEE Conference on Engineering in Medicine and Biological Systems*, pp. 4009-4012, 2004.

[10] C. Vargas-Irwin, J.P. Donoghue, "Automated spike sorting using density grid contour clustering and subtractive waveform decomposition," *Journal of Neuroscience Methods*, vol. 164, no. 1, pp. 1-18, 2007.

[11] J. Dai, et al. "Experimental study on neuronal spike sorting methods," *IEEE Future Generation Communication Networks Conference*, pp. 230-233, 2008.

[12] R.J. Vogelstein, K. Murari, P.H. Thakur, G. Cauwenberghs, S. Chakrabartty, C. Diehl, "Spike sorting with support vector machines", *Proceedings of Annual International Conference on IEEE Engineering in Medicine and Biology Society*, pp. 546-549, 2004.

[13] K.H. Kim, S.S. Kim, S.J. Kim, "Advantage of support vector machine for neural spike train decoding under spike sorting errors", *Proceedings of Annual International Conference on IEEE Engineering in Medicine and Biology Society*, pp. 5280-5283, 2005.

[14] R. Boostani, B. Graimann, M.H. Moradi, G. Pfurtscheller, "A comparison approach toward finding the best feature and classifier in cue-based BCI", *Medical and Biological Engineering Computing*, vol. 45, no. 403-412, 2007.

[15] G. Zouridakis, D.C. Tam, "Identification of reliable spike templates in multi-unit extracellular recordings using fuzzy clustering," *Computer Methods and Programs in Biomedicine*, vol. 61, no. 2, pp. 91-98, 2000.

[16] B. Gosselin, "Recent advances in neural recording microsystems," *Sensors*, vol. 11, no. 5, pp. 4572-4597, 2011.

[17] T. Chen, Z. Yang , W. Liu, L. Chen, "NEUSORT 2.0: A multiple-channel neural signal processor with systolic array buffer and channel-interleaving processing schedule," *Proceedings of Annual International Conference on IEEE Engineering in Medicine and Biology Society*, pp. 5029-5032, 2008.

[18] E. Shih, J. Guttag, "Reducing energy consumption of multi-channel mobile medical monitoring algorithms," *Proceedings of International Workshop on Systems and Networking Support for Healthcare and Assisted Living Environments*, no. 15, pp. 1-7, 2008.

[19] R.E. Schapire, "A brief introduction to boosting," *Proceedings of International Joint Conference on Artificial Intelligence*, pp. 1401-1406, 1999.

[20] B. Schölkopf, A.J. Smola, *Learning with kernels - support vector machines, regularization, optimization and beyond*, Cambridge, MA: The MIT Press, 2002.

[21] C.-W. Hsu, C.-J. Lin, "A comparison of methods for multi-class support vector machines," *IEEE Transactions on Neural Networks*, vol. 13, pp. 415-425, 2002.

[22] O. Mangasarian, D. Musicant, "Successive overrelaxation for support vector machines," *IEEE Transactions on Neural Networks*, vol. 10, no. 5, pp. 1032-1037, 1999.

[23] C.-W. Hsu, C.-J. Lin, "A simple decomposition method for support vector machines," *Machine Learning*, vol. 46, pp. 291-314, 2002.

[24] V.N. Vapnik, *Statistical learning theory*, John Wiley & Sons, 1998.

[25] V. Franc, V. Hlavac, "Multi-class support vector machine", *Proceedings of IEEE International Conference on Pattern Recognition*, vol. 2, pp. 236-239, 2002.

[26] J. Platt, *Fast training of support vector machines using sequential minimal optimization*, in Advances in kernel methods: Support vector learning, chapter, Cambridge, MA: The MIT Press, 1999.

[27] R.O. Duda, P.E. Hart, D.G. Stork, *Pattern classification*, John Wiley & Sons, 2000.

[28] B. Scholkopf, P. Knirsch, C. Smola, A. Burges, Fast approximation of support vector kernel expansions, and an interpretation of clustering as approximation in feature spaces, in P. Levi, M. Schanz, R.J. Ahler, F. May, editors, *Mustererkennung 1998-20,* pp. 124-132, Berlin, Germany, 1998. Springer-Verlag.

[29] H. Lee, S.-Y. Kung, N. Verma, "Improving kernel-energy tradeoffs for machine learning in implantable and wearable biomedical applications," *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 1597-1600, 2011.

[30] K.-H. Lee, N. Verma, "A low-power processor with configurable embedded machine-learning accelerators for high-order and adaptive analysis of medical-sensor signals," *IEEE Journal of Solid-State Circuits*, vol. 48, no. 7, pp. 1625-1637, 2013.

[31] C.J. Burges, "Simplified support vector decision rules," *International Conference on Machine Learning*, pp. 71-77, 1996.

[32] S.R.M. Ratsch, T. Vetter, "Efficient face detection by a cascaded support vector machine expansion," *A Royal Society of London Proceedings Series*, vol. 460, pp. 3283-3297, 2004.

[33] P. Koch, B. Bischl, O. Flasch, T. Bartz-Beilstein, W. Konen, "On the tuning and evolution of support vector kernels," *Evolutionary Intelligence*, vol. 5, pp. 153-170, 2012.