# Bayes Optimal Template Matching for Spike Sorting – Combining Fisher Discriminant Analysis with Optimal Filtering

Felix Franke1,2, ?Quian Quiroga?, Andreas Hierlemann1, Klaus Obermayer2

(1) ETH Zürich, Department of Biosystems Science and Engineering, 4058 Basel, Switzerland  
(2) Technische Universität Berlin, School for Electrical Engineering and Computer Science, Berlin, Germany

## Abstract

Spike sorting, i.e. the estimation of single neuronal spike trains from extracellular measurements, is a crucial but often error prone step in the analysis of neuronal firing behavior. Usually, three different problems have to be solved: the detection of spikes in the extracellular recordings, the estimation of the number of neurons and their prototypical template spike waveforms, and the assignment of individual spikes to those putative neurons. If the template spike waveforms are assumed to be known, template matching can be used to solve the detection and classification problem. Here, we show that for the colored Gaussian noise case and arbitrary number of recording channels, the optimal template matching is given by a form of optimal linear filtering and can be derived via linear discriminant analysis. This gives a Bayesian interpretation to the well known optimal matched filter output. The usual problem of setting the right detection threshold vanishes, since the optimal threshold can be computed analytically. The method can be implemented by a simple linear filter bank derived from the templates and could be used for real-time spike sorting but might be applicable in general to detection and classification problems of transient signals. Its application significantly decreases the error rate on two publicly available benchmark data sets in respect to state-of-the-art template matching procedures that are used in commercial products.

## Introduction

The detection and classification of voltage deflections in extracellular recordings - called spikes - caused by action potentials of neurons is known as spike sorting. It is necessary if single neuronal activities must be resolved from multi-neuronal firing activity. The assignment of spikes to neurons is only possible because the morphology of the neuron, its spatial position in respect to the recording electrode(s) and the surrounding medium cause the recorded waveform of every neuron to be distinct. Furthermore, spikes from the same neuron have a highly similar waveform. It is, therefore, possible to cluster the extracellular potentials into groups based on their waveform. The assumption is that all spikes within a group were emitted by the same neuron. Many studies focus on the problem of finding similarity measures and clustering procedures that are most suited for spike sorting (e.g. (Lewicki, 1998; Letelier and Weber, 2000; Shoham and Fellows, 2003; Quiroga et al., 2004)).

The recorded data consists of two different linearly added components, noise and spikes. Therefore, for any given piece of data, spike sorting must solve three problems: First, pieces of data that contain only noise must be separated from pieces that contain both, spikes and noise. This is called the spike detection problem. Second, spikes must be grouped into clusters of similar spikes that originate supposedly from the same neuron, the so called clustering problem. And third, each spike must be assigned to one neuron. This is called the classification problem. At first the classification problem seems to be redundant since the clustering problem already needed to assign spikes to putative neurons. But there are several reasons why it is important to keep the classification conceptually separated from the clustering: Classification is usually much faster than clustering. In an online application, e.g., it might be desirable to use a fast classifier that was derived from an initial offline clustering for real-time spike sorting (see Fig. 1). Additionally, many clustering procedures scale not very well with the number of spikes and their application becomes infeasible for very long recordings. Then, only a subset of spikes could be clustered and the rest simply classified. And last but not least some clustering algorithms explicitly make use of a classifier during the clustering to assign spikes to neurons.

One way to build such a classifier is to calculate the average of all elements for each cluster. This cluster center is called template. A classifier that is purely based on templates is often referred to as template matching (TM). Each unclassified spike is compared to each template and is subsequently assigned to the template that was most similar to it according to some appropriate similarity measure.

Here, we focus exclusively on the detection and the classification problem: If the number of neurons and their templates are known, what is the optimal way to perform spike sorting? Different approaches to solve this problem were proposed (Lewicki, 1998; Letelier and Weber, 2000) including filter based methods (Roberts and Hartline, 1975) and methods that perform TM to directly compare the templates to the data (see e.g. (Keehn, 1966; Friedman, 1968) but also recent approaches (Zhang et al., 2004; Vargas-Irwin and Donoghue, 2007)). Although TM is of great importance to extracellular physiology, it was not thoroughly investigated yet what the optimal strategy is. Even current commercial products rely on very simple strategies like Euclidean distance (ED) (Plexon Inc, 2009; Cambridge Electronic Design Limited, 2012) with manual threshold selection.

We show that although the ED and the Mahalanobis distance (MD) are well suited for classification of perfectly aligned spikes, they are not suited for spike detection and TM of spikes whose waveform is affected by noise, registration jitter and other spikes and which are thus hard to correctly align.

We proceed by showing that TM and filtering are directly related and propose a finite impulse response (FIR) filter based online spike sorting procedure that is - under the additional assumption of Gaussian noise - optimal in a Bayesian sense. We derive its optimality from Fisher's linear discriminant analysis (LDA) and show that the outputs of optimal matched filters can be interpreted in a Bayesian sense. The proposed Bayes optimal template matching (BOTM) computes the linear discriminant functions by a convolution of FIR filters with extracellular data and solves both the detection and the classification problem in an optimal way. In contrast to the other methods BOTM provides the optimal detection threshold analytically and does not require manual intervention. Using BOTM to simultaneously detect and classify spikes removes the need for spike alignment since the peak of the detector output is a robust estimate of the true position of the spike.

We evaluate our method on two previously published benchmark data sets and show that we can significantly improve template matching performance in both cases. Additionally, using BOTM as a post-processing step after clustering can reduce the number of errors during the initial spike sorting step.

Due to its computational simplicity BOTM is especially useful for online and real-time implementation in the context of closed-loop experiments after the initial templates have been found (Obeid et al., 2004; Rutishauser and Schuman, 2006; Franke et al., 2010; Einevoll et al., 2012).

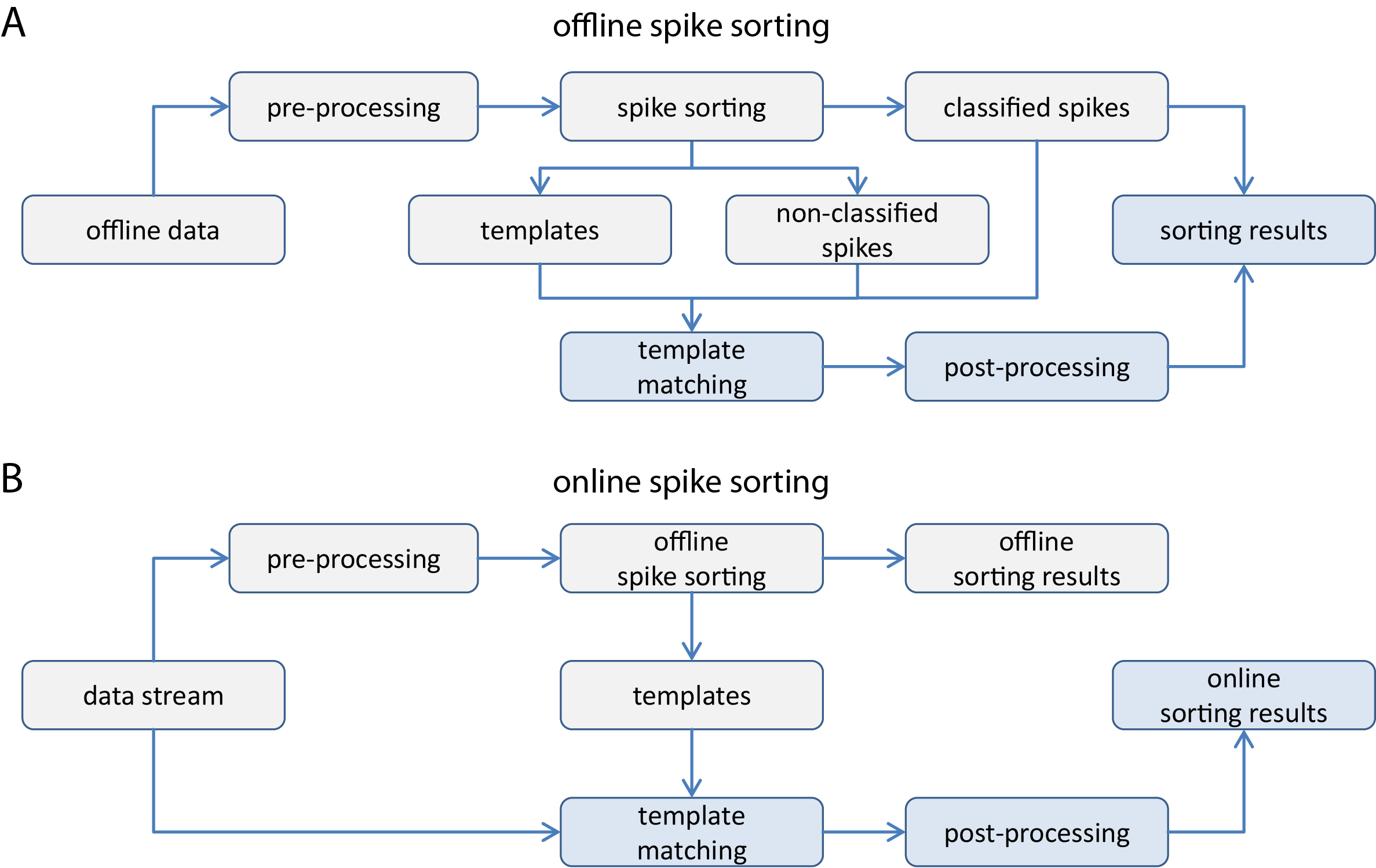
****

Fig. 1 Processing steps for template matching (TM) based spike sorting. In two different cases TM can be used for spike sorting. A In the offline case, data is sorted first with a spike sorting technique. The resulting templates can be used to either reassign already classified spikes (dotted line) or to assign spikes that could not be successfully clustered, e.g., due to an overlap of two different spikes. B TM can also be used for online spike sorting. Here, templates are estimated in an offline learning phase and then used to classify spikes online. In both, the online and offline case, templates can also be used to detect spikes in the data. This work only discusses the processing steps marked in blue.

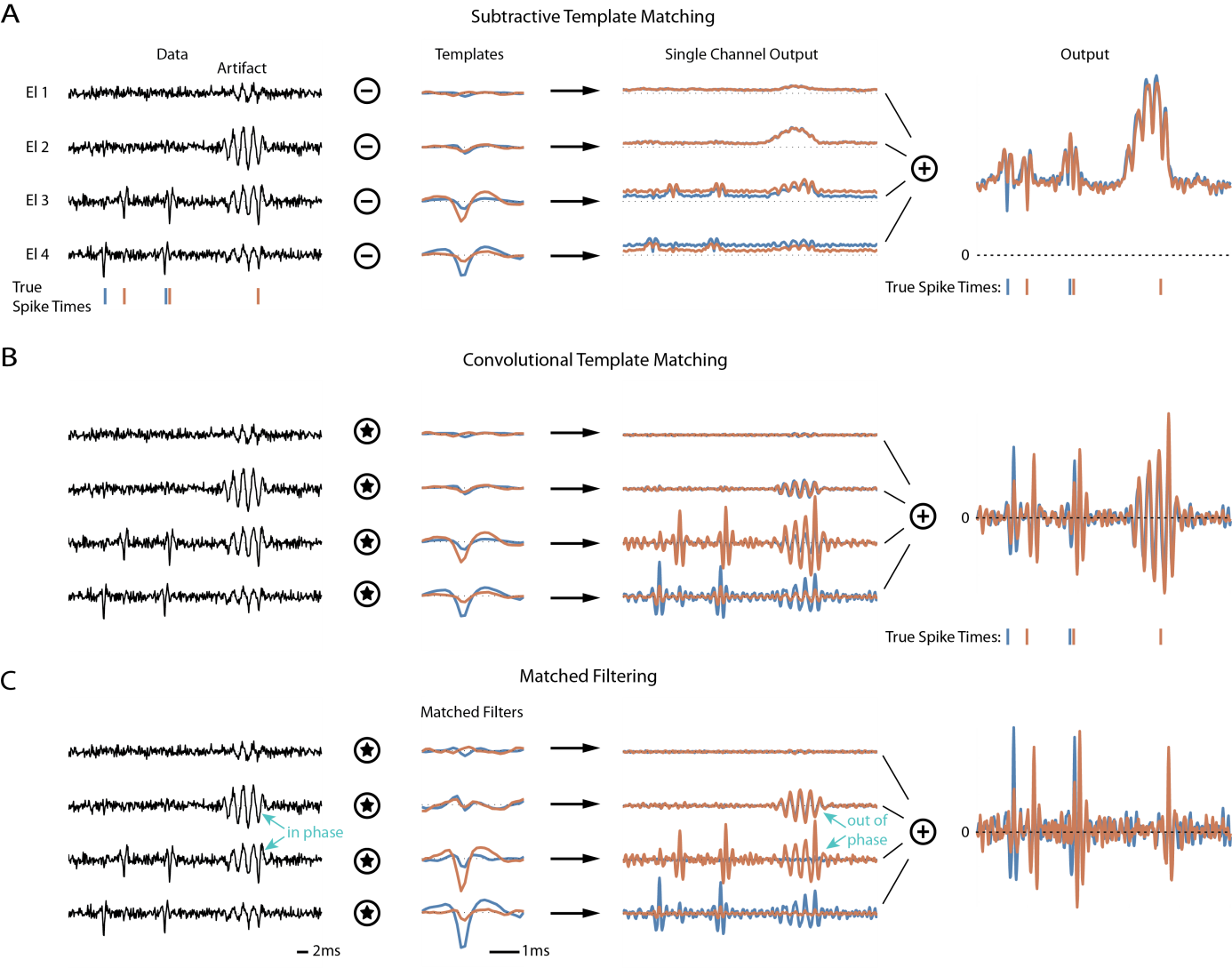


Fig. 2 Comparison of different TM techniques in a toy example (artificial data; y-axis arbitrary units). A short piece of a 4 electrode recording was simulated by copying two templates and a simulated artifact into white Gaussian noise. The first column of plots shows the data, the second column the templates (or the derived filters), the third column the respective single channel TM outputs and the last column shows the final TM output. A Known templates are subtracted at each possible position from the data for each channel individually and the norm of the residual is computed. This is the Euclidean distance (ED) of a piece of data to each template. B The templates are convolved with the data. Convolution has the advantage that also overlapping spikes produce detectable peaks in the output. However, an artifact will also lead to strong responses. C If the noise and artifact characteristics are known, matched filters can be used for TM which optimally enhances the response to the templates. Interestingly, in this toy example, the artifact cancellation is achieved by using the artifact on the seemingly useless electrode 2 - on which the templates have nearly no energy - to cancel the artifact on channels 3 and 4 (marked with “in phase” and “out of phase” respectively). In principle, matched filtering can suppress the effect of artifacts while still resolving overlapping spikes to a certain degree. All three methods have the problem that setting the detection threshold is, due to crosstalk between the filter outputs for different neurons, not straight-forward.

## Method

This section introduces different methods for TM and explains the involved concepts. More formal proves of the methods can be found in the appendix.

### Definitions

Spike sorting relies on the assumption that the action potentials of a single neuron lead to extracellular recorded spikes with the same waveform (Lewicki, 1998). This is generally not true, since spikes of the very same neuron are known to vary, e.g., dependent on the recent firing history of the neuron (Fee et al., 1996) in bursts and slowly over time (Franke et al., 2010). In this work we will assume non-varying waveforms but address the issue of different forms of spike waveform. We define the template of neuron as where is the waveform for neuron on channel and is the vector transpose. The single channel waveforms are samples long. Thus is a vector with dimensions, where is the number of recording channels and single channel waveforms are concatenated. We use the analog definition for recorded data: where represents samples of recorded data around time (see Fig. 2 for a graphical explanation). This way the output of a multichannel FIR filter applied to the recordings (which is usually denoted as where is the convolution) can be elegantly expressed in terms of a vector multiplication: . The noise covariance matrix (Pouzat et al., 2002) is then given by: with being a piece of data where no spikes are present (we assume that the data is high pass filtered and thus ).

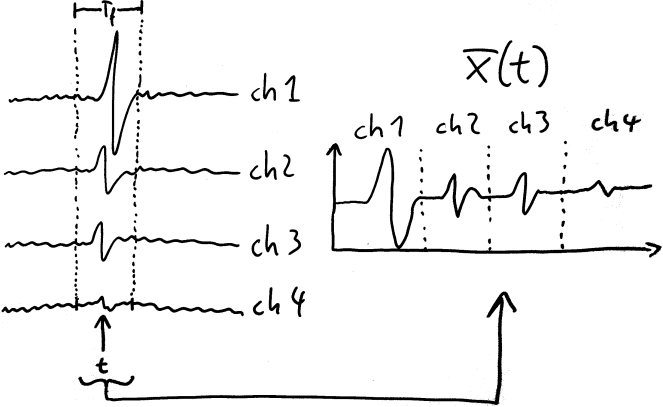


Fig. 3 Illustration of the definition of mathematical entities. Pieces of data of length on all channels are concatenated to form a vector .

### Classical Template Matching

Two different similarity measures between a piece of data and a template are commonly employed for TM: Euclidean distance TM (e.g., (Sato et al., 2007; Plexon Inc, 2009; Cambridge Electronic Design Limited, 2012)) and convolution or cross-correlation TM (Friedman, 1968; Kim and McNames, 2007). The ED at time between data and template is defined as

and the convolution as

.

If the noise covariance matrix is known, the recorded data can be prewhitened (Pouzat et al., 2002; Rutishauser and Schuman, 2006) before matching the templates. This is equivalent to using the Mahalanobis distance instead of the ED and matched filtering instead of the convolution:

Fig. 2 shows a comparison of different TM techniques on a simple toy example. But theoretical considerations alone can immediately show that neither of the techniques is well suited to solve both, the detection and classification problem.

### Detection performance

Assume that is a piece of recorded data containing noise only while contains noise and template . The performance of a detector depends on the overlap of the distributions and . For and these distributions are given by

|  |  |  |
| --- | --- | --- |
| Euclidean Distance |  |  |
| and noise: |  |  |
| noise only: |  |  |

|  |  |  |
| --- | --- | --- |
| Convolution |  |  |
| and noise: |  |  |
| noise only: |  |  |

follows a distribution with degrees of freedom. The respective distributions start to overlap for and , even for a relatively small number of samples and electrodes. This effect gets worse with the number of samples per template and recording electrodes. Zero padding all templates or adding electrodes on which all templates are zero will severely reduce detection performance. Despite the fact that the added dimensions are useless for detecting the presence of the templates, noise in those dimensions will enter the detector output.

On the other hand, and do not suffer from that problem since they do not include the term: Noise is projected on a one dimensional subspace spanned by the template (or filter). If an electrode is added on which the template is zero, this electrode will be completely ignored by and (might still use this electrode for noise suppression, see Fig. 2).

### Classification performance

For classification, however, for which only the relative distance of a piece of data and the templates is important, the term of DTM is irrelevant, since it is the same for all templates. Thus, it has no influence to which template a piece of data is assigned, as long as we are sure that the piece actually contains a spike.

In contract, for the classification case, it is straight-forward to construct a scenario in which and fail. Consider the case where one template is nearly a scaled version of another template . This directly yields . The templates are not anymore distinguishable by thresholding the output of a linear filtering. The same problem occurs if two templates - which have their peaks on the same electrode set - also have high amplitude differences, a situation that is very likely to occur in real recordings. Therefore, purely convolutional methods together with a simple threshold are not suitable as classifiers.

### Bayes optimal template matching (BOTM)

The previous considerations suggest using or for spike detection while using or for spike classification. However, they can be combined in an optimal way using a Bayesian approach. The Bayes optimal template matcher (BOTM) provides the best possible detection and classification performance (given that the assumptions of Gaussian noise and constant templates are not too strongly violated) and is given by:

where is the prior probability to observe a spike of neuron . For a derivation see the appendix and (Franke, 2011). BOTM computes matched filter outputs for each template and shifts them along the y-axis by a constant. The additive constant depends on the respective templates energy and its probability to occur in the data, which is related to the neurons firing rate. Additionally, BOTM is the only technique that directly provides the optimal detection threshold (see appendix). Fig. 4 shows an illustrative comparison of BOTM and other TMs on a spike sorting benchmark (Quiroga et al., 2004).

### Influence of amplitude variability

So far we assumed that a neuron always produces a spike in form of its template. However, this is not always true. Especially, spikes that follow briefly after another spike, tend to have a changed waveform (Fee et al., 1996). The change seems to be mostly a linear scaling of the whole waveform with a minor nonlinear component. This raises the question how do the different TMs react to amplitude variability of the spikes? Consider neuron producing a spike that is scaled by factor : . For and this will result in an increase of the template matching output by making it significantly more difficult to detect the spike. On the other hand, classification is only affected if the waveform change actually brings the spike closer to the template of another neuron (see section on classification performance).

For and , linear amplitude change causes a linear decrease of the detector outputs for all templates. It is not possible that the change will affect the classification of the spike. Thus, as long as the spike can be detected, it will be always assigned to the same template. Classification with and is thus invariant to linear amplitude scaling. The downside of this is that is makes it impossible to distinguish templates that are linearly dependent.

The detection performance of the BOTM does not suffer from the decrease in detection performance like or but is still able to distinguish between templates that are linearly scaled. Thus, amplitude variability could influence the classification. However, as shown in the result section, this seems to be no problem for the benchmark data sets used in this study.

### Noise Covariance Matrix

The noise covariance plays a crucial role for the MD, the matched filter and BOTM as well as for some spike sorting procedures (Pouzat et al., 2002; Rutishauser and Schuman, 2006). In all cases its inverse is needed. However, this matrix can be badly conditioned, i.e., it might have eigenvalues that are close to zero, which makes its inversion instable. A standard procedure to invert ill defined covariance matrices is diagonal loading or shrinkage []. Here, a target covariance matrix , often the identity, is added or merged to the original covariance matrix. We decided to chose the diagonal of as the target and blended it with the original matrix: . We chose but the exact choice of this parameter did not affect the results.

## B_manual.png

Fig. 4 Example of different TM techniques on a public benchmark data set with a single electrode simulation. A Raw simulated data of two different data files from the benchmark. The true spike times (shown as ticks below the data) are known since the data was simulated. At the position of the true spike times the respective templates are superimposed over the data. B Output of Euclidean distance TM. Even on the high signal to noise ration data file (left) not all spikes can be reliably detected by thresholding the output: the first spike of the green neuron is obscured by the partial overlap. On the low SNR file (right) the output cannot be thresholded in a useful way to detect spikes. C Convolution-based TM gives in both cases clear peaks at the positions of the spikes. But not always is the highest peak also from the output of the correct template. D The matched filter is in this case very close to the sonvolution-based TM. But all spikes would be classified as the blue neuron by a simple threshold. E Thresholding the BOTM output is the only method that yields all spike times and their correct identities in both examples (with the exception of the overlapping spike in the difficult data set). This overlap would have to be resolved by a post-processing step. Since the color of the highest peak is not always clearly visible, a line with the same color is placed above the respective peaks.

### Evaluation metrics

We evaluated the performance of the described template matching procedures on two different data sets described below. Both were preprocessed in a similar way: The available ground truth information was used to cut true spikes from the recording. This avoided the problem of aligning the spikes. The template matchers were evaluated for detection and classification performance separately.

For the detection task the TM output distribution was computed for all spikes as well as for pieces of noise. Then, the overlap of noise and spike distribution was computed and, except for the BOTM, the optimal threshold which minimized the detection error estimated numerically. The alternative would have been to compare receiver operator characteristic curves, but the BOTM provides the optimal threshold analytically. Therefore, we decided to compare its performance to the optimal performance of the other methods (note that this overestimates the performance of the other methods since in a real situation, their optimal threshold is not given). We used a prior probability of to express the fact that in real data any given piece of data is more likely to contain noise rather than a spike. The results do not depend strongly on the value of . We explored values from 0.5 to 0.9999 and always found qualitatively the same results. A value of 0.99 (i.e., we assumed it 100 time more likely to observe noise than a spike) was used for the final evaluation.

Detection Performance was defined as the percentage of true spikes found at the optimal threshold.

The classification task depended on the data set and is described below.

### Benchmark 1 (Q): Evaluation on simulated data with full ground truth

The proposed template matching was evaluated on the publicly available spike sorting benchmark data set described in [4] named Benchmark 1 (Q). The data set consists of 4 sub benchmarks labeled “Easy1” to “Diffcult2”. Every sub benchmark consists of 4 different data files (with the exception of “Easy1” which has 8) with decreasing signal to noise ratios. All files contain a simulated single channel extracellular recording with 24kHz sampling rate and 3 simulated neurons. Templates and the noise covariance matrix were calculated using the available ground truth information. Short periods of simulated data of length samples starting 15 samples before the given spike time points were cut and averaged to create the templates. The templates were than aligned on their peak. The template matching outputs were computed on the true spike waveforms. Each spike was assigned to the template with the maximal response. Spikes that were correctly assigned were counted as TP while spikes assigned to the wrong template were counted as classification errors CL. We used the following quantities for the performance comparison:

### Benchmark 2 (H): Evaluation on real recording with partial ground truth

This data set is the part of the hc-1 data set described in (Henze et al., 2000) and publicly available under <http://crcns.org/>. The following files were used: d11221.002, d11222.001, d12821.001, d14521.001 and d533101. Each data file consists of a several minute simultaneous intra- and extracellular recording in rat hippocampus. Ground truth information is available for only one single neuron and needs to be extracted from the respective intracellular recording. We chose the files depending on the quality of the intracellular recording. The extracellular recordings were high pass filtered at 300Hz.

Spike sorting using mean-shift clustering (Marre et al., 2012) was performed in the space of the first 6 principle components after prewhitening to estimate the templates. The sorting was not optimized manually. The template of the cluster best matching to the ground truth was used to estimate the performance of the template matching procedures and will be referred to as target template. For all data files the target template was very similar to the template that can be obtained by using only the spikes given by the ground truth, i.e., in all cases one of the clusters matched the ground truth well enough to get good template estimation.

For the classification task, the template matching output was computed for all templates and all spikes of the ground truth neuron as well as for all other spikes detected during the spike sorting. Spikes of the ground truth that were correctly matched to the target template were counted as true positives (TP), those assigned to other neurons as false negatives (FN). Spikes not included in the ground truth that were assigned to the target template were counted as false positives (FP) those assigned to other templates as true negatives (TN). The following quantities were used to estimate performance:

To compare the performance of BOTM to the spike sorting performance of reported in (Quiroga et al., 2004), we did not use the ground truth to cut perfectly aligned spikes. Instead, BOTM was run on the whole data and spikes were detected in the TM output. Overlaps were partly resolved by the narrow peaks of the filter outputs but we also employed SIC (see appendix).

## Results

The results are shown in Fig. 6 and confirm the previous theoretical considerations and are consistent between data sets. While and give nearly 100% detection performance in all cases, and have a very bad detection performance. For classification and have a consistently high performance while and perform poorly. The high standard deviation of and can be explained by the fact that on data files where the target template is one of the largest templates in the data, its classification by and is easily possible knowing the correct detection threshold. In all cases BOTM is at least as good as the other methods and outperforms them on average by decreasing the average error by around 90% in respect to Euclidean distance template matching. We compared the performance using different noise covariance matrices, the raw matrix, only its diagonal and the loaded matrix. Using the diagonal only performed poorly on average while diagonal loading always increased performance significantly (results not shown) but this effect was stronger on benchmark 2 possibly because of the higher number of dimensions due to the multielectrode nature of the data. In all cases and BOTM had the highest classification (and rejection) performance while using and BOTM had the best detection performance. This shows that noise is clearly correlated and taking this covariance structure into account improves spike sorting results, both for the detection and classification problems. Furthermore, since BOTM can be used to detect and classify spikes at the same time, there is no need to align the spike waveforms. This is crucial for classification performance. If spikes are misaligned due to noise or overlapping spikes, also the distance based TMs classification performance will be decreased.

Tab. 1 compares the performance of BOTM to the clustering and detection performance of the spike sorter “wave\_clus” presented in (Quiroga et al., 2004). It has to be noted, however, that this comparison is not totally fair. “wave\_clus” is a spike sorting algorithm that has to detect spikes and cluster them without knowing the templates. This is a harder problem than TM with given templates. But it shows that using BOTM after initial spike sorting on the whole data could strongly decrease detection and classification errors and has thus a potential application as a post-processing technique for all spike sorting algorithms.

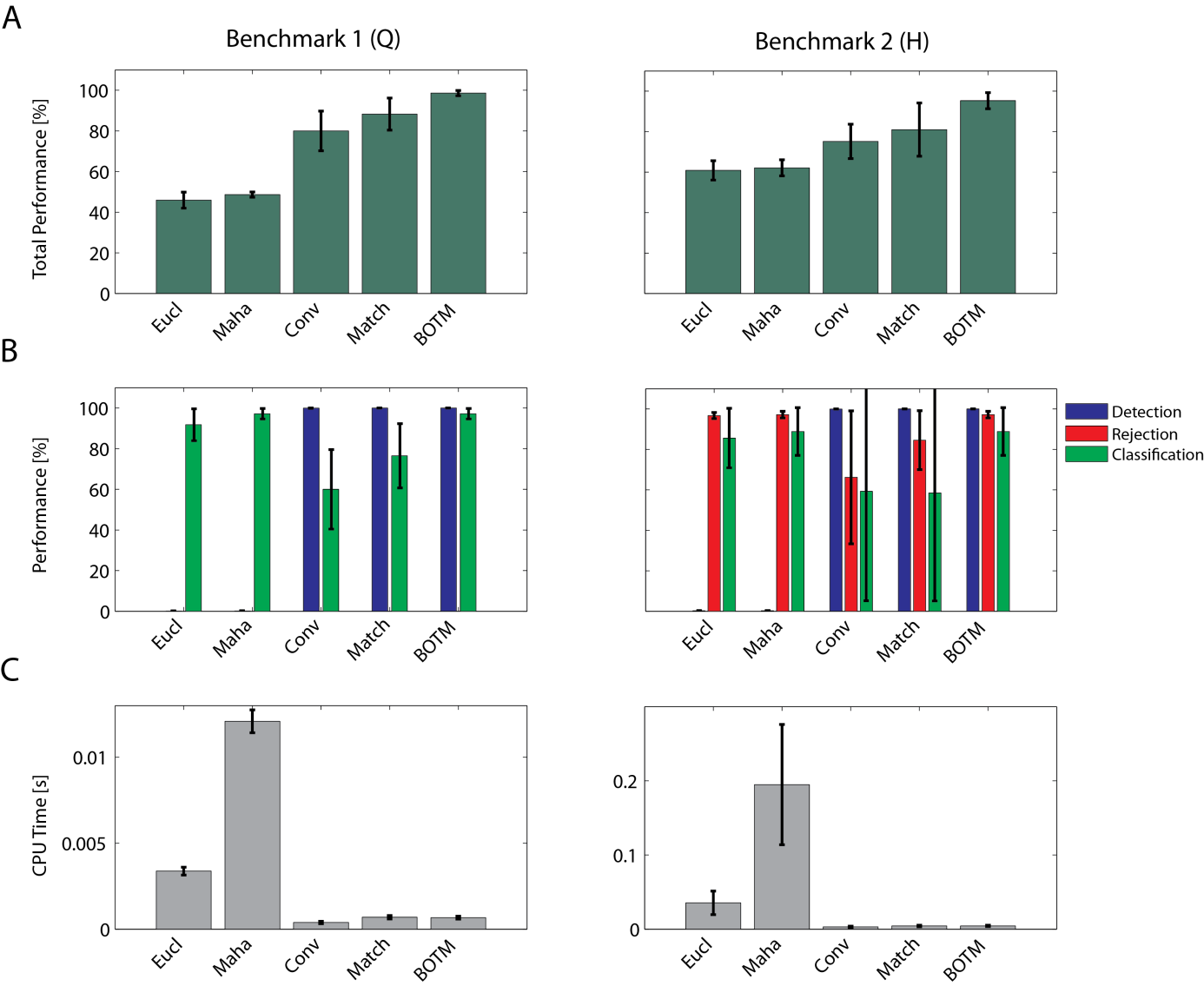
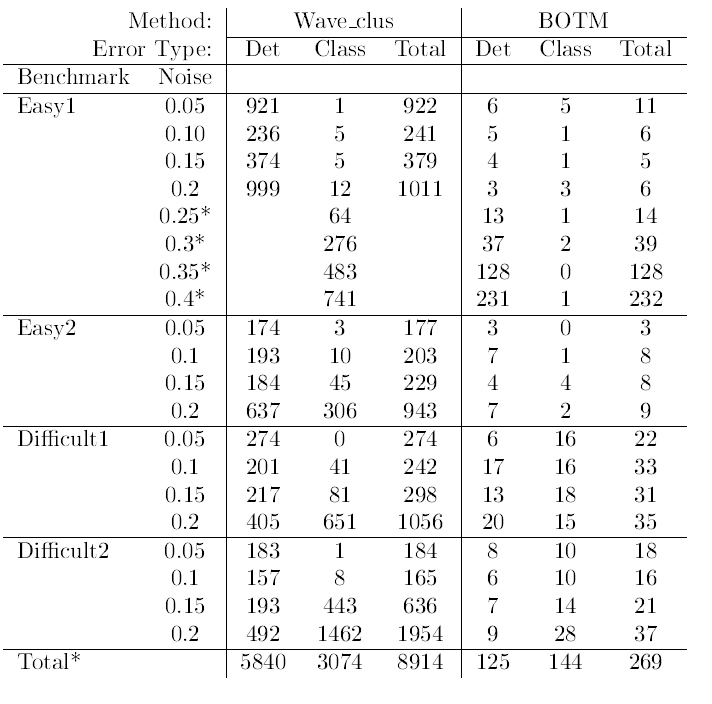


Fig. 6 Performance evaluation on two public benchmark data sets “Q” (left column) and “H” (right column). A Total performance of different TMs. BOTM outperforms the other methods consistently over files and benchmarks. B Performance for different error categories. The two distance based TM algorithms “Eucl” and “Maha” perform poorly in the detection task while the convolution-based TM “Conv” and “Match” perform poorly on the classification task. Only BOTM can solve both problems at the same time while providing the same classification performance as. C Average computation time per template and benchmark file. The convolution-based methods and BOTM are considerably faster than the distance based methods.



Tab. 1 Performance comparison BOTM vs. Wave\_clus. Spike sorting errors for the proposed method BOTM and the method described in (Quiroga et al., 2004) “Wave\_clus”. “Det” are detection errors, “Class” classification errors and “Total” their sum. For “Wave\_clus” classification was done independently of detection. For classification all spikes were used also those that were not detected. Detection results for high noise levels of Easy1 were not reported. Column sums were computed ignoring noise levels 0.25 to 0.4 of Easy1 marked by an \*. Note that the comparison is not fair since “Wave\_clus” is a full spike sorter while for BOTM the templates were assumed to be known (details see text).

## Discussion

We analyzed the performance of different template matching procedures and showed that distance based TM and convolution-based TM both suffer strongly from orthogonal problems: distance based TM is not suitable to detect spikes while convolution-based TM might be very bad in classifying spikes depending on the templates in the data. For the colored Gaussian noise case we derived the optimal TM with a Bayesian approach and showed its performance on two benchmark data sets which provides a link between Fisher Discriminant analysis and optimal filtering. The proposed BOTM algorithm outperforms the other methods, is computationally very efficient and provides the optimal detection threshold analytically. The BOTM procedure is an ideal candidate for TM. Its computational simplicity makes it suitable for a hardware implementation. The method can in principle be applied for all detection and classification problems of multiple transient signals.

### Noise Covariance matrix and Gaussian noise assumption

The presented method explicitly assumes that noise is multivariate colored Gaussian, i.e., with noise covariance matrix . But there are indications that noise in extracellular recordings does not strictly follow a multivariate Gaussian distribution (Shoham and Fellows, 2003). Neither of the data sets used in this study were constructed to follow this assumption. In fact, the noise in the benchmark data set from (Quiroga et al., 2004) was created by copying many templates with small amplitudes into the data and the data set from (Henze et al., 2000) are real recordings where noise is likely to contain small amplitude spikes from neurons that are farer away from the electrodes. However, in the data sets we analyzed so far, using the Gaussian noise assumption always significantly increased spike detection and classification performance.

### Bursts

The problem of bursting neurons was not thoroughly addressed in this study. During a burst a neuron fires many spikes at a high rate. Individual spikes during a bursts tend to vary in amplitude (Fee et al., 1996) and probably also shape (although to what degree was, especially for multielectrode recordings, not sufficiently quantified yet). To overcome the problem that spikes during a deviate from the template of the respective neuron, several templates could be used per neuron and the maximal TM output chosen. If the burst induced waveform change is mainly scaling, it would not be necessary to construct new filters. The filter outputs could be derived directly from the “mother” template by scaling and addition of a constant. How well this approach works on real data, however, is hard to estimate since reliable ground truth information is not easily available.

Hardware implementation

The computationally most expensive part of BOTM is the convolution of the FIR filters with the data. FIR filters are very well suited for a parallelized hardware implementation. However, a possible hardware implementation would have to be very flexible since the number of filters per electrode is highly dependent on the position of neurons and will be different for each recording. Furthermore, to save resources, the number of electrodes per neuron should be limited to the least possible number while still maintaining high sorting performance. How to optimally choose which and how many electrodes to use for each neuron still needs to be investigated.

## Acknowledgements

We would like to thank Robin Ince, Christophe Pouzat for helpful discussions and Wendelin Böhmer and Espen Hagen for feedback on an early draft of this paper. This work was supported by DFG GRK 1589/1 and the German Federal Ministry of Education and Research (BMBF) with the grants 01GQ0743 and 01GQ0410 and by the European Community through the ERC Advanced Grant 267351, “NeuroCMOS”. Felix Franke acknowledges individual support through an EU-funded Marie Curie Training Network of FP6: CT 2006-035854, CELLCHECK. Part of this work was already published in (Franke, 2011).

## Appendix

### Matched filtering

Missing, keep short, only give basic principle of NOTCH filter.

### Derivation of Linear Discriminant Analysis

Consider the classical multi-class classification problem: given a set of data points each belonging to one class and a new unlabeled data point . To which class should we assign ? If we want to minimize the classification error, we will assign to the class with maximal conditional probability, i.e. assign to class if

|  |  |  |
| --- | --- | --- |
|  | . | (6.1) |

Since the conditional probability is usually hard to compute it is easier to use Bayes rule and calculate the likelihood instead:

|  |  |  |
| --- | --- | --- |
|  |  | (6.2) |

In the context of the classification problem we are not interested in the exact probabilities but only in the relations between them. We are searching for a discriminant function that will give us the same classificator as (eq. 6.1)

|  |  |  |
| --- | --- | --- |
|  |  | (6.3) |

but omits all unnecessary computations; e.g. the denominator in (6.2) appears on both sides of the inequality and can thus be ignored:

|  |  |  |
| --- | --- | --- |
|  |  | (6.4) |

Further assuming that is multivariate normally distributed with -dimensional mean and covariance matrix we get

|  |  |  |
| --- | --- | --- |
|  | *.* | (6.5) |

The term is again shared by all and can be dropped. Taking the logarithm, the discriminant function can be reformulated to (note that we had to multiply the equation by -2, thus now needs to be smaller than all )

which is called Fisher's quadratic discriminant function. If we can constrain all classes to share the same covariance matrix (homoscedasticity), can be simplified further to

|  |  |  |
| --- | --- | --- |
|  |  | (6.6) |

yielding the linear discriminant function used in linear discriminant analysis (LDA) (Fisher, 1936). The result is similar to the M-ary detection case in (Choi, 2010), chapter 2.5.

In the context of spike sorting the classes represent different neurons and all vectors are assumed to be perfectly aligned spike waveforms. Then the assumption that all clusters have the same covariance matrix is partly justified: noise in the spike waveforms can be modeled as being independent of the identity of the spike.

### Derivation of Bayes Optimal Template Matching (BOTM)

We are now combining the results from the LDA with matched filtering. According to the generative model commonly assumed for extracellular data, spike templates are centers of multivariate normal distributions with a shared (noise) covariance matrix. If we now interpret each sample in the original data as a possible position for a spike of any neuron we need to compute the discriminant function of all samples to all templates.

Interestingly, the linear discriminant functions can be calculated by a linear filtering with the matched filter:

|  |  |  |
| --- | --- | --- |
|  |  | (6.7) |

is a vector representing the period of multi-channel data starting at time t. is the optimal matched filter for template. The term does not depend on and can be treated as constant .

|  |  |  |
| --- | --- | --- |
|  |  | (6.7) |

This way the filter output of the matched filters (shifted on the y-axis by a constant ) can directly be interpreted as the Fisher discriminant function.

### Discriminant function for noise

Obviously, not every sample of the data belongs to a spike. The problem to decide whether a given sample is noise instead of a spike can be solved by introducing a "template" (and thus a discriminant function) for noise. Noise was assumed to have a mean of zero. Taking the null vector 0 as just another template n, the corresponding discriminant function for noise will be

|  |  |  |
| --- | --- | --- |
|  |  | (6.8) |

with being the prior probability for a given sample of data to be noise. is obviously a constant and is the inverse probability to observe a spike:

|  |  |  |
| --- | --- | --- |
|  |  | (6.9) |

Thus eq. (6.7) provides a natural threshold for spike detection.

### Discriminant function for overlapping spikes

The discriminant function for an overlap between spikes from neuron and with time shift can be defined analogously by introducing a new template for that overlap

|  |  |  |
| --- | --- | --- |
|  |  | (6.10) |

where denotes the template of neuron shifted by time :

|  |  |  |
| --- | --- | --- |
|  |  | (6.11) |

Eq. (6.11) can be reformulated to

|  |  |  |
| --- | --- | --- |
|  |  | (6.12) |

with being a time shifted version of . Eq. 6.12 allows computing the overlap discriminant function without actually constructing the overlap template or computing its filter. This can be done analogously for overlaps with more than 2 participating spikes.

### Derivation of Subtractive Interference cancellation

If the waveforms - a set of optimal matched filters is tuned to - overlap in the time domain, as may be the case with extracellular spikes, the corresponding filter responses can overlap as well. To allow for the simultaneous detection of multiple waveforms, subtractive interference cancellation (SIC) (Moshavi, 1996) is used. This means once the presence of a waveform has been detected, it will be subtracted from the data and the responses of the other filters will be updated. For this subtraction the precise position of the waveform in the data is needed, otherwise a shifted version might be subtracted, causing a residual error. The original sampling rate of the data might not be fine enough; therefore, it can be advantageous to upsample the data before subtraction. To save computation time, SIC can be performed directly on the discriminant functions.

Assume template was found at time in the data. To perform SIC we need to remove the influence of this spike from the data and update all discriminant functions so that they correspond to the respective discriminant function for overlaps between and but after removal of spike . This can be achieved by updating all discriminant functions according to

|  |  |
| --- | --- |
|  | (6.13) |

which directly updates to the correct value of after theoretical subtraction of from the data at time .

## References

Cambridge Electronic Design Limited (2012). Spike 2 Version 7 Manual, page 15-8, point 3. Available at: http://www.ced.co.uk/img/Spike7.pdf.

Choi, J. (2010). *Optimal Combining and Detection*. Cambridge University Press.

Einevoll, G. T., Franke, F., Hagen, E., Pouzat, C., and Harris, K. D. (2012). Towards reliable spike-train recordings from thousands of neurons with multielectrodes. *Current opinion in neurobiology* 22, 11–17. Available at: http://www.ncbi.nlm.nih.gov/pubmed/22023727 [Accessed November 8, 2012].

Fee, M. S., Mitra, P. P., and Kleinfeld, D. (1996). Variability of extracellular spike waveforms of cortical neurons. *Journal of neurophysiology* 76, 3823–33. Available at: http://www.ncbi.nlm.nih.gov/pubmed/8985880.

Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of Eugenics* 7, 179–188.

Franke, F. (2011). *Real-Time Analysis of Extracellular Multielectrode Recordings*. Berlin: PhD Thesis, Technische Universität Berlin Available at: http://opus.kobv.de/tuberlin/volltexte/2012/3387/.

Franke, F., Natora, M., Boucsein, C., Munk, M. H. J., and Obermayer, K. (2010). An online spike detection and spike classification algorithm capable of instantaneous resolution of overlapping spikes. *Journal of Computational Neuroscience* 29, 127–148. Available at: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2950077&tool=pmcentrez&rendertype=abstract.

Friedman, D. H. (1968). *Detection of Singals by Template Matching*. Baltimore, Maryland: Johns Hopkins Press.

Henze, D. A., Borhegyi, Z., Csicsvari, J., Mamiya, A., Harris, K. D., and Buzsáki, G. (2000). Intracellular features predicted by extracellular recordings in the hippocampus in vivo. *Journal of neurophysiology* 84, 390–400. Available at: http://www.ncbi.nlm.nih.gov/pubmed/10899213 [Accessed April 30, 2013].

Keehn, D. G. (1966). An Iterative Spike Separation Technique. *Biomedical Engineering, IEEE Transactions on* BME-13, 19–28.

Kim, S., and McNames, J. (2007). Automatic spike detection based on adaptive template matching for extracellular neural recordings. *J Neurosci Methods* 165, 165–174. Available at: http://dx.doi.org/10.1016/j.jneumeth.2007.05.033.

Letelier, J. C., and Weber, P. P. (2000). Spike sorting based on discrete wavelet transform coefficients. *Journal of neuroscience methods* 101, 93–106. Available at: http://www.sciencedirect.com/science/article/pii/S0165027000002508 [Accessed August 14, 2011].

Lewicki, M. S. (1998). A review of methods for spike sorting: the detection and classification of neural action potentials. *Network (Bristol, England)* 9, R53–78. Available at: http://www.ncbi.nlm.nih.gov/pubmed/10221571.

Marre, O., Amodei, D., Deshmukh, N., Sadeghi, K., Soo, F., Holy, T. E., and Berry, M. J. (2012). Mapping a Complete Neural Population in the Retina. *Journal of Neuroscience* 32, 14859–14873. Available at: http://www.jneurosci.org/cgi/doi/10.1523/JNEUROSCI.0723-12.2012 [Accessed October 24, 2012].

Moshavi, S. (1996). Multi-user detection for DS-CDMA communications. *IEEE Communications Magazine* 34:1010, 124–136.

Obeid, I., Nicolelis, M. a. ., and Wolf, P. D. (2004). A multichannel telemetry system for single unit neural recordings. *Journal of Neuroscience Methods* 133, 33–38. Available at: http://linkinghub.elsevier.com/retrieve/pii/S0165027003003212 [Accessed August 1, 2012].

Plexon Inc (2009). Plexon Offline Spike Sorter Manual, Ch. 5.6.2. Available at: http://www.plexon.com/sites/default/files/Offline Sorter v3 Manual.pdf.

Pouzat, C., Mazor, O., and Laurent, G. (2002). Using noise signature to optimize spike-sorting and to assess neuronal classification quality. *Journal of neuroscience methods* 122, 43–57. Available at: http://www.ncbi.nlm.nih.gov/pubmed/12535763.

Quiroga, R. Q., Nadasdy, Z., and Ben-Shaul, Y. (2004). Unsupervised spike detection and sorting with wavelets and superparamagnetic clustering. *Neural computation* 16, 1661–87. Available at: http://www.ncbi.nlm.nih.gov/pubmed/15228749.

Roberts, W. M., and Hartline, D. K. (1975). Separation of multi-unit nerve impulse trains by a multi-channel linear filter algorithm. *Brain research* 94, 141–9. Available at: http://www.ncbi.nlm.nih.gov/pubmed/1148860 [Accessed August 26, 2011].

Rutishauser, U., and Schuman, E. (2006). Online detection and sorting of extracellularly recorded action potentials in human medial temporal lobe recordings, in vivo. *Journal of neuroscience* 154, 204–224. Available at: http://www.sciencedirect.com/science/article/pii/S0165027006000033 [Accessed August 14, 2011].

Sato, T., Suzuki, T., and Mabuchi, K. (2007). Fast automatic template matching for spike sorting based on Davies-Bouldin validation indices. *Conf Proc IEEE Eng Med Biol Soc* 2007, 3200–3203. Available at: http://dx.doi.org/10.1109/IEMBS.2007.4353010.

Shoham, S., and Fellows, M. (2003). Robust, automatic spike sorting using mixtures of multivariate t-distributions. *Journal of neuroscience methods* 172, 112–121. Available at: http://www.sciencedirect.com/science/article/pii/S016502700800246X [Accessed August 14, 2011].

Vargas-Irwin, C., and Donoghue, J. P. (2007). Automated spike sorting using density grid contour clustering and subtractive waveform decomposition. *J Neurosci Methods* 164, 1–18. Available at: http://dx.doi.org/10.1016/j.jneumeth.2007.03.025.

Zhang, P.-M., Wu, J.-Y., Zhou, Y., Liang, P.-J., and Yuan, J.-Q. (2004). Spike sorting based on automatic template reconstruction with a partial solution to the overlapping problem. *Journal of neuroscience methods* 135, 55–65. Available at: http://www.ncbi.nlm.nih.gov/pubmed/15020089.