ELSEVIER

Journal of Neuroscience Methods 157 (2006) 303–316

JOURNAL OF NEUROSCIENCE METHODS

www.elsevier.com/locate/jneumeth

Spike sorting: Bayesian clustering of non-stationary data

Aharon Bar-Hillel [a,*], Adam Spiro [b,1], Eran Stark [c,2]

[a] *The Interdisciplinary Center for Neural Computation, The Hebrew University of Jerusalem, Jerusalem 91904, Israel*
[b] *School of Computer Science and Engineering, The Hebrew University of Jerusalem, Jerusalem 91904, Israel*
[c] *Department of Physiology, Hadassah Medical School, The Hebrew University of Jerusalem, Jerusalem 91120, Israel*

Received 5 September 2005; received in revised form 19 March 2006; accepted 23 April 2006

Abstract

Spike sorting involves clustering spikes recorded by a micro-electrode according to the source neurons. It is a complicated task, which requires much human labor, in part due to the non-stationary nature of the data. We propose to automate the clustering process in a Bayesian framework, with the source neurons modeled as a non-stationary mixture-of-Gaussians. At a first search stage, the data are divided into short time frames, and candidate descriptions of the data as mixtures-of-Gaussians are computed for each frame separately. At a second stage, transition probabilities between candidate mixtures are computed, and a globally optimal clustering solution is found as the maximum-a-posteriori solution of the resulting probabilistic model. The transition probabilities are computed using local stationarity assumptions, and are based on a Gaussian version of the Jensen–Shannon divergence. We employ synthetically generated spike data to illustrate the method and show that it outperforms other spike sorting methods in a non-stationary scenario. We then use real spike data and find high agreement of the method with expert human sorters in two modes of operation: a fully unsupervised and a semi-supervised mode. Thus, this method differs from other methods in two aspects: its ability to account for non-stationary data, and its close to human performance.
© 2006 Elsevier B.V. All rights reserved.

Keywords: Clustering; Jensen–Shannon divergence; Mixture of Gaussians; Monkey recordings; Non-stationary data; Semi-supervised learning; Spike sorting

1. Introduction

In extra-cellular recording of brain activity, a micro-electrode usually picks up the activity of multiple neurons. Spike sorting is the task of finding a clustering of the spike data such that each cluster is comprised of the spikes generated by a different neuron. Currently, this task is mostly done manually. It is a tedious mission, requiring hours of human work for a single recording session. Many algorithms were proposed in order to automate this process (see Lewicki, 1998 for a review) and some were implemented to provide a helping tool for manual sorting (Harris et al., 2000; Lewicki, 1994; Yu and Kreiman, 2000). However, the ability of suggested algorithms to replace the human worker has been quite limited. One of the main obstacles for a successful spike sorting application is the non-stationary nature of the data (Lewicki, 1998). The primary source of this non-stationarity is slight movements of the recording electrode. Small drifts of the electrode's location, which are almost inevitable, result in changes of the shapes of recorded spikes over time. Other sources of non-stationarity are variable background noise, and, perhaps, changes of the characteristic spike shape generated by a neuron. The increasing usage of multiple electrode recordings turns non-stationarity into an important issue, since electrodes are being placed in a single location for long durations.

Manual sorting often uses the first two principal component analysis (PCA) coefficients (PCs) to represent spike data (see Hulata et al., 2002; Quiroga et al., 2004 for other, wavelet-based representations). Such a representation preserves up to 93% of the variance of the recorded spikes (Abeles and Goldstein, 1977). A human often clusters spikes by visual inspection of the projected spikes in small time frames, in which non-stationary effects are insignificant. Problematic scenarios which can appear due to non-stationarity are demonstrated in Fig. 1 and include: (1) Movements and considerable shape changes of clusters over time. Such movement causes a cluster to be smeared when observed at a large time window. (2) Two clusters which are initially well-separated may move until they merge into one. A split of a single cluster is possible in the same manner. We would

* Corresponding author. Tel.: +972 2 658 4079; fax: +972 3 535 7155.
E-mail addresses: aharonbh@cs.huji.ac.il (A. Bar-Hillel), adams@cs.huji.ac.il (A. Spiro), eranstark@md.huji.ac.il (E. Stark).
[1] Tel.: +972 3 5238584.
[2] Tel.: +972 2 6758381.

0165-0270/$ – see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.jneumeth.2006.04.023

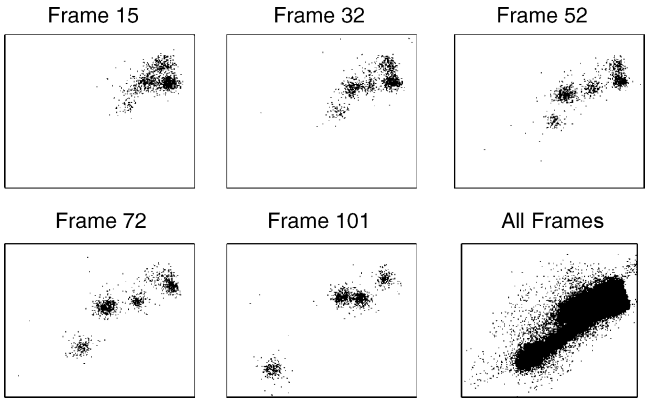


Fig. 1. Example of several time frames from a single electrode recording. Five time frames from a sequence of 116 are presented with their chronological frame numbers (1000 spikes per frame). The bottom right panel shows all the data points in a single frame (115,782 spikes). Spikes were projected on two principal components. In this recording, several clusters move constantly. In the process, a big fuzzy cluster splits into distinguishable clusters and later, two of those become indistinguishable again. While single time frames may have several plausible clustering interpretations, the sequence as a whole is usually less ambiguous. Clearly, the "right" clustering cannot be seen when all data points are observed in a single frame.

like to distinguish between different clusters whenever possible, and to state that a cluster is a multi-unit cluster when separation is not possible anymore.

Most spike sorting algorithms do not address the presented difficulties at all, as they assume full stationarity of the data. Some methods do try to cope with the lack of stationarity by grouping data points into many small clusters and then identifying clusters that can be combined to represent the activity of a single unit. For instance, Fee et al. (1996) use inter-spike interval (ISI) information and density of points on the borders between clusters to decide which clusters should be combined. Snider and Bonds (1998) point out that an ISI criterion is not valid for correlated neurons and base the cluster unification decision on the density of points between clusters. Shoham et al. (2003) suggest handling non-stationary clusters by modeling clusters using a *t*-distribution. This distribution can partially describe asymmetric smeared clusters. None of these methods partition the data into time frames and hence none can cope with complicated cases requiring this information. For example, clusters of similar spike shapes existing at distant time frames are always unified by such methods, while tracking the cluster through time may reveal that it is not the same cluster.

Emondi et al. (2004) suggested a semi-automated method in which the data are divided into small time frames. Each time frame is clustered manually, and then the correspondence between clusters in consecutive time frames is established automatically. In order to establish the correspondence, match scores between clusters are computed using a heuristic metric, and a greedy procedure is then used to choose matching pairs. No splits or merges of clusters are considered in that method.

In this paper we suggest a fully automated method for solving the clustering problem offline in a Bayesian framework. A preliminary version of the method was presented in a conference paper (Bar-Hillel et al., 2005). Trying to mimic human experts, we focus here on two-dimensional spike representations, although our algorithm can be applied to higher dimensions as well. In brief, the spike data are divided into short time frames in which stationarity is a reasonable assumption. At a first stage, good descriptions of the data as mixtures-of-Gaussians are computed for each time frame independently. Many possible mixture descriptions are considered for each time frame. In a second stage, transition probabilities between consecutive mixtures are computed, and the complete dataset is modeled as a chain of local mixture models. The final clustering, as well as the correspondence between clusters in consecutive time frames, is found as the maximum-a-posteriori (MAP) solution of the chain model. This global optimization allows the algorithm to successfully disambiguate problematic time frames in a way similar to the behavior of a human expert. The model and its optimization are presented in Section 2.

An important contribution to the success of the method is made by the derivation and computation of the transition probabilities. The assumptions and algorithms used to derive these probabilities are described in Section 3. In essence, we assume that the data are approximately stationary in two consecutive time frames, and we look for a hidden mixture model that explains well the Gaussian components in both time frames. We frame the search for the "right" hidden mixture model as a constrained optimization problem with a Jensen–Shannon (JS) based score. The problem can be optimally solved if the number of clusters is assumed to be constant, and we suggest an efficient suboptimal strategy in a general case in which splits and merges of clusters are allowed.

Empirical validation of the proposed method is presented in Section 4. We first test the performance of the method using synthetically generated spike data. In this case we know the correct source neuron for each spike, and hence we can verify the correctness of sorting in various scenarios and compare our method with other recently proposed spike sorting methods. The agreement between the clustering of the proposed method and the correct labels is found to be higher than that of other methods in most scenarios. Second, we test the performance of the method operating in an unsupervised mode using electrode recordings from premotor cortices of Macaque monkeys (Stark et al., 2006). The clustering is evaluated by computing its agreement with manual clustering done by a human expert, and high agreement rates are found in most cases. In another experiment, we compare the agreement between the algorithm's and human clustering solutions to the agreement among several expert human sorters. The algorithm is comparable to the human experts in this test. Finally, we test the algorithm in a semi-supervised mode, in which a human user clusters part of the time frames, and the algorithm has to fill in the rest. In this mode, the clustering results highly agree with the intentions of the human expert, while human labor is substantially reduced.

2. A chain of Gaussian mixtures

A single recording session may contain tens or hundreds of thousands of spikes recorded over several hours (Section 4). While spike clusters are not necessarily stationary over such

periods, it is reasonable to assume stationarity for shorter periods of several minutes. Note that we assume stationarity of spike *shapes* and not of any temporal firing characteristics. In that case it is known (Lewicki, 1998) that the cluster density is well approximated by a Gaussian with a general covariance matrix. We therefore approach the problem by a two stage process (the flow is summarized in Algorithm 1). In the first stage, the spike data are divided into time frames of fixed duration or a fixed number of spikes, N (in most experiments we used $N = 1000$). We then look for a set of likely mixture-of-Gaussians solutions for each time frame independently. This is a search stage, heuristic in nature, and its purpose is to find good *local* solution candidates which constrain the search space in the second, *global* stage. In this second stage the best combination of local solutions is found by computing the MAP solution in a global generative model of the complete data. These two algorithmic stages are the core of the algorithm and they are described in Sections 2.1 and 2.2, respectively. In Section 2.3 we describe how repetition of local and global optimization stages in a feedback loop is used to refine the solution. Another performance improvement is gained by introducing some rule-based post-processing, described in Section 2.4.

Algorithm 1. Spike sorting

1. *Divide the data into T non-overlapping time frames*.
2. *Search for local solutions*: for each $t \in \{1,\ldots, T\}$ search for M^t candidate mixtures using EM.
   For $i = 1,\ldots, N_f$ do
   3. *Mixing of solutions*: for each $t \in \{1,\ldots, T\}$, import and adapt selected solutions from neighboring frames $[t-k,\ldots, t+k]$.
   4. *Calculate transition scores*: for each $t \in \{1,\ldots, T-1\}$, calculate transition scores between all candidates of time frame t and candidates of frame $t+1$.
   5. *Find a global solution*: calculate the MAP path using the Viterbi algorithm.
   If $i < N_f$, use the found path to re-initialize the local candidates pool.
6. *Post-processing*: identify special clusters.

2.1. Local solutions

Denote the observable spike data by $D = \{d\}$, where each spike $d \in R^2$ is described by the vector of its PC coefficients. The data are divided into T disjoint groups, $\{D = \{d_i^t\}_{i=1}^{N_t}\}_{t=1}^{T}$. We assume that in each frame the data are approximated well by a mixture-of-Gaussians, where each Gaussian corresponds to a single source neuron. Each Gaussian in the mixture may have a different covariance matrix. Since the number of components in the mixture is not known a-priori, it is assumed to be within a certain range per time frame (we used 1–6 in our experiments).

In the search stage, an expectation-maximization (EM) algorithm (Dempster et al., 1977) is used to find a set of M^t mixture-of-Gaussians candidates for each time frame t. The EM is run with different values for the "number of clusters" parameter and different initial conditions. This creates the basic pool of solutions in each time frame. Then, for each time frame t, the best mixture solutions found in the neighboring time frames $[t-k, \ldots, t+k]$ are imported into the local solutions pool (we used $k = 2$). These solutions are also adapted by using them as the initial conditions for the EM and running a low number of EM rounds. This mixing of solutions between time frames is repeated several times (twice in our experiments), so good local solutions are propagated backward and forward in time. Finally, the solution list in each time frame is pruned to remove similar solutions.

In order to handle outliers, which may be background spikes or non-spike events, each candidate mixture contains an additional "background model" Gaussian. The parameters of this model are set to $\mu = 0$, $\Sigma = K\Sigma_t$, where Σ_t is the covariance matrix of the data in frame t, and $K > 1$ is a constant. During the EM process only the weight of this Gaussian in the mixture is allowed to change. The flexibility of this weight allows the EM to adapt to variable degrees of background noise.

When working in a semi-supervised mode, human guidance is naturally incorporated at this stage. The human supervisor provides local solutions to a subset $S \subset \{1,\ldots, T\}$ of the T time frames. For each frame $t \in S$, the manual clustering is translated into a mixture-of-Gaussians and considered as the only candidate solution of time frame t. This constrains the algorithm to choose the manual clustering in the appropriate frames and enables the spreading of these solutions into the neighboring frames through the "Mixing of solutions" stage of Algorithm 1.

2.2. A global model

After the search stage, each time frame t has a candidate list of M^t mixture models $\{\Theta_i^t\}_{t=1,i=1}^{T,\ M^t}$. Each mixture model is described by a triplet $\{\alpha_{i,l}^t, \mu_{i,l}^t, \Sigma_{i,l}^t\}_{i=1}^{K_{i,t}}$, denoting the Gaussians' weights, means, and covariance matrices, respectively. Given these candidate models we define a discrete random vector $Z = \{z^t\}_{t=1}^{T}$ in which each component z^t has a value range of $\{1, 2, \ldots, M^t\}$. "$z^t = j$" has the semantics of "at time frame t, the data are distributed according to the candidate mixture Θ_j^t". In addition we define for each spike d_i^t a hidden discrete "label" random variable l_i^t. This label indicates which Gaussian in the local mixture hypothesis is the source of the spike. Denote by $L^t = \{l_i^t\}_{i=1}^{N^t}$ the vector of labels of time frame t and by L the vector of all labels.

We describe the probabilistic relations between D, L, and Z using a Bayesian network with the structure described in Fig. 2. Using the network structure and assuming independent

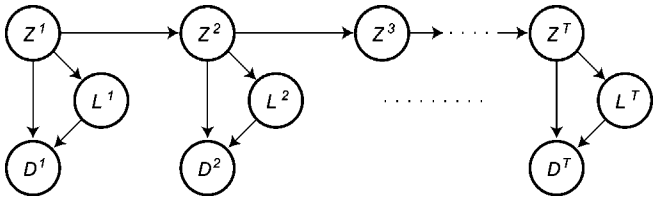


Fig. 2. A Bayesian network model of the data generation process. The network has a hidden Markov model (HMM) structure, but unlike HMM it does not have fixed states and transition probabilities over time. The variables and the conditional probability distributions are explained in Section 2.2.

and identically distributed samples, the joint probability decomposes into

$$\begin{aligned} P(Z, L, D) &= P(Z)P(L|Z)P(D|L, Z) \\ &= P(z^1)\prod_{t=2}^{T} P(z^t|z^{t-1})\prod_{t=1}^{T} P(L^t|z^t)P(D^t|L^t, z^t) \\ &= P(z^1)\prod_{t=2}^{T} P(z^t|z^{t-1})\prod_{t=1}^{T}\prod_{i=1}^{N^t} P(l_i^t|z^t)P(d_i^t|l_i^t, z^t) \quad (1) \end{aligned}$$

The complete log-likelihood is therefore given by

$$\begin{aligned} \log P(Z, L, D) &= \log P(z^1) + \sum_{t=2}^{T} \log P(z^t|z^{t-1}) \\ &\quad + \sum_{t=1}^{T}\sum_{i=1}^{N^t} [\log P(l_i^t|z^t) + \log P(d_i^t|l_i^t, z^t)] \quad (2) \end{aligned}$$

and we wish to maximize this log-likelihood over all possible choices of the hidden variables L, Z. Notice that by maximizing the probability of both data and labels we avoid a tendency to prefer mixtures with many Gaussians, a tendency which appears when maximizing the probability for the data alone. The conditional probability distributions (CPDs) of the points' labels and the points themselves, given an assignment to Z, are

$$\log P(l_k^t = j|z^t = i) = \log \alpha_{i,j}^t$$

$$\begin{aligned} \log P(d_k^t|l_k^t = j, z^t = i) &= -\frac{1}{2}[n \log 2\pi + \log \left|\Sigma_{i,j}^t\right| \\ &\quad + (d_k^t - \mu_{i,j}^t)^t \Sigma_{i,j}^{t-1}(d_k^t - \mu_{i,j}^t)] \quad (3) \end{aligned}$$

For the prior of the first frame, $P(z^1)$, we use a uniform CPD. The transition probabilities $P(z^t = j|z^{t-1} = i)$ are computed based on the assumption that the visible mixtures Θ_i^{t-1} and Θ_j^t arise from a single mixture model with unknown parameters. The computation relies on a Gaussian version of the Jensen–Shannon divergence, and it includes the establishment of a correspondence mapping between clusters in the two frames. These ideas are formalized and described in Section 3.

Given the transition probabilities, we can conduct exact inference in the chain model. The MAP solution for the model is found using the Viterbi algorithm (Mackay, 2003). Labels are then unified using the correspondences established between the chosen mixtures in consecutive time frames. When the correspondence is not one-to-one, as in the cases of merges and splits, labels are not unified. Instead, the merged cluster is identified as a multi-unit.

2.3. A feedback loop

The initial pool of solutions computed at the local search stage is very diverse, but important local solutions may still be missing. This in turn damages the quality of the optimal chain found in the global stage. Once we have found a good global chain candidate using the Viterbi algorithm, we can use it to guide the search for better local candidates. It is hence reasonable to create another, more "focused", pool of solutions and repeat the global optimization stage.

In order to create such a focused candidate pool the initial pool of each time frame is set to contain a single solution, the one chosen by the global optimization. Then, the solutions mixing stage is repeated. The new candidate pool of time frame t thus contains only solutions that appeared in nearby time frames in the global path and their adaptations. In addition, solutions with clusters that appear in the good global chain for only small fraction of the time frames (1/10 in our implementation) are dropped from the candidate list. The new "search space" for the global optimization is now smaller. However, it contains a rich collection of variations over the best path found so far. This enables a delicate refinement of the path and improves the final solution. A single repetition of the feedback loop is usually enough to obtain the desired result, so we use $N_f = 2$ in Algorithm 1.

2.4. Post-processing

Using only the stages described so far, the algorithm gives reasonable results. However, in order to reach performance level of a human expert, additional processing is required. The algorithm mimics human treatment of non-stationarity well, but the human expert supplements this basic logic with some considerations of additional important problems. These problems arise when assumptions of the standard clustering analysis fail. One such problem, of "sparse" clusters, is described in Fig. 3. In this case the sample size (number of spikes) of certain clusters is extremely small, and they cannot be identified in small time frames. A second problem is the existence of extremely non-stationary events, caused for instance by a large sudden movement of the electrode. To cope with such phenomena we added a post-processing stage to the algorithm.

Sparse clusters tentatively represent spikes of neurons with very low firing rates. Such clusters cannot be detected within individual small time frames, since such a frame contains only a few spikes of this cluster that are labeled as background. In order to detect sparse clusters, the entire dataset is considered in a single time frame and is resampled to reduce the weight of dense areas. Several descriptions of the resampled data as mixtures-of-Gaussians are computed, using various numbers of clusters. For each cluster in each description the distances to other clusters in the mixture are computed using a Gaussian Jensen–Shannon

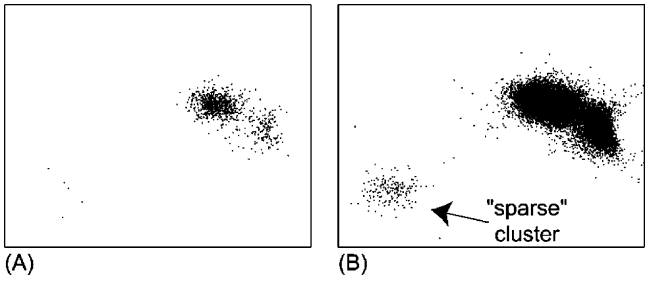


Fig. 3. A "sparse" cluster, attributed to a neuron with an extremely low firing rate. (A) A single time frame. Due to its sparseness, the cluster in the lower left corner cannot be detected in small time windows. (B) Viewing the entire dataset at once allows the cluster to be clearly observed.

score (see Section 3). A candidate for a sparse cluster is then the cluster for which the distance to the nearest cluster in its mixture is maximal. The candidate is declared a sparse cluster if most (more than 75%) of its points were classified as background spikes in the main algorithm. This process enables us to locate most of the sparse clusters (about 90%) found by humans with almost no spurious detections.

Abrupt large movements of the recording electrode or other extreme conditions produce spike recordings that are extremely non-stationary. In such cases even our modest local stationarity assumptions do not hold, and the correspondence established between clusters in consecutive time frames is not reliable. Human sorters usually drop such data. Setting thresholds on the transition probabilities computed by the algorithm can identify these cases. We use two such thresholds in series. Transitions with score (Eq. (11)) lower than the first threshold are identified as "not-continuous" (the value of this threshold was set manually, based on visual judgment). For each dataset, the length of the longest sequence of continuous transitions is computed. If this length is below a second threshold (six frames in our implementation), the dataset is marked as "extremely non-stationary". Less than 10% of the recordings can be classified as extremely non-stationary, and indeed the above classification method manages to correctly identify them in the test data. The results reported in Section 4 do not include such extremely non-stationary data.

3. Transition probabilities

The transition CPDs of the form $P(z^t|z^{t-1})$ are based on the assumption that the distributions of the Gaussian sources are approximately stationary in pairs of consecutive time frames. While spikes in two consecutive frames are labeled using two different "visible" mixture descriptions, they are assumed to be generated by a single, hidden, mixture-of-Gaussians with unknown parameters. We further assume that each Gaussian component in the two mixtures arises from a single Gaussian component of the hidden mixture. The hidden mixture thus induces a partition of the set of visible components into clusters, where Gaussian components in the same cluster are assumed to arise from the same hidden source. Our estimate of $P(z^t = t|z^{t-1} = i)$ is based on the probability of seeing two large samples labeled according to Θ_i^{t-1} and Θ_j^t under the assumption of a single hidden joint mixture.

We formalize the relevant events and the notion of a hidden mixture model in Section 3.1. In Section 3.2, we establish a connection between the transition probability and a known quantity in the statistics literature, the Jensen–Shannon (JS) divergence. We define a specialization of this quantity for the Gaussian case, and cast the transition probability computation as an optimization of a Gaussian JS-based score over the possible hidden mixture models. If the family of allowed hidden mixture models is not further constrained, the optimization problem derived in Section 3.2 is trivially solved by choosing the most detailed partition, where each visible Gaussian component has its own hidden source. This happens because a richer partition, which does not merge many Gaussians, gets a higher score. In Section 3.3 we suggest natural constraints on the family of allowed partitions in the two cases of constant and variable number of clusters through time, and present algorithms for both cases.

3.1. Problem formulation

A probabilistic account of transitions between mixtures-of-Gaussians requires definitions of the relevant events and exact formulation of the probabilistic model used to assess these events. We suggest to define the relevant events in terms of the labeled samples induced by the Gaussian mixtures. A labeled sample $S = (D, L) = \{d_i, l_i\}_{i=1}^N$ is a set of points labeled according to their source, where $l_i \in \{1, \ldots, K\}$ gives the label of point d_i, K being the number of sources. A mixture model for time frame t induces a labeled sample by labeling each spike according to its most probable source. Different mixture candidates of the same time frame induce different labeled samples. In Cover and Thomas (1991) and in Csiszar (1998), a useful method for handling large samples of discrete variables is presented, entitled "theory of types". We extend the concepts of "type" and "type class" introduced in that theory to labeled samples. The most general form of this extension is beyond the scope of this paper (see Bar-Hillel and Spiro, 2005). Here, we briefly state the basic notions required for the spike sorting application.

For our purposes, the "type" or "empirical distribution" of a labeled sample is the mixture-of-Gaussians implied by the maximum likelihood (ML) estimation from the sample. Formally, for a labeled sample $S = (D, L)$ it is the mixture $\hat{\Theta}(S) = \{\alpha_q, \mu_q, \Sigma_q\}_{q=1}^K$, where

$$\alpha_q = \frac{|\{j : l_j = q\}|}{N}, \qquad \mu_q = \frac{1}{|\{j : l_j = q\}|} \sum_{\{j:l_j=q\}} d_j,$$

$$\Sigma_q = \frac{1}{|\{j : l_j = q\}|} \sum_{\{j:l_j=q\}} (d_j - \mu_q)(d_j - \mu_q)^t \tag{4}$$

and $|\{j{:}l_j = q\}|$ is the number of points that have the label q. The power of the "type" concept arises from the fact that the likelihood of a sample depends, under any Gaussian mixture model, only on the sample's type. Therefore, interesting events in the sample space are sets of samples of a similar type. For a mixture model Θ we define the "type class", $\mathrm{TC}_{N,\varepsilon}(\Theta)$, as

$$\mathrm{TC}_{N,\varepsilon}(\Theta) = \{S : \|\hat{\Theta}(S) - \Theta\|_\infty < \varepsilon, |S| = N\} \tag{5}$$

where $|S|$ is the number of points in the sample S. The norm $\|\cdot\|_\infty$ demands that all the parameters of the mixtures will be ε-close to each other. This definition gives us the basic events we wish to measure (see Eqs. (6) and (7) below).

Assume that in two consecutive time frames we observed two N-point labeled samples, $S^1 = (D^1, L^1)$ and $S^2 = (D^2, L^2)$. The joint sample $S = S^1 \cup S^2$ is a labeled sample with $K^1 + K^2$ possible labels. We order this set of $K^1 + K^2$ Gaussians and refer to them using one index. If we assume the data are approximately stationary in the two frames, then $K^1 \approx K^2$ and the data are actually generated from a hidden mixture Θ^H with $K^H \approx K^1$

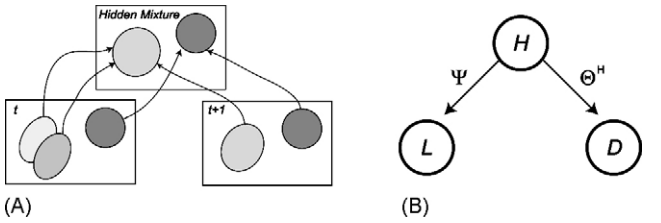


Fig. 4. (A) A diagram of the correspondence R between components of the hidden and visible mixtures. The diagram illustrates a "merge" event of the two bottom-left clusters. (B) A Bayesian network representation of the relations between the data and the hidden labels. The visible labels L and the sampled data points D are independent given the hidden labels H.

components. The visible labels in this case contain irrelevant "noise", since points from the same hidden source are labeled with different visible labels in S^1 and in S^2. We formalize this relation between visible labels and hidden source labels using a correspondence mapping R:$\{1, \ldots, K^1 + K^2\} \to \{1, \ldots, K^H\}$ which matches each Gaussian component in the visible mixtures to its hidden source component in Θ^H. An example can be seen in Fig. 4A.

Denote the labels of the sample points under the hidden mixture $H = \{h_i^j\}_{i=1}^{N^j}$, $j = 1, 2$. The values of these variables are given by $h_i^j = R(l_i^j)$, where l_i^j is the visible label index. The probabilistic dependence between a data point, its visible label, and its hidden label is illustrated by the Bayesian network model in Fig. 4B. We assume that a data point is obtained by choosing a hidden label and then sample the point from the relevant hidden component. The visible label is then sampled based on the hidden label using a multinomial distribution with parameters $\Psi = \{\psi_q\}_{q=1}^{K^1+K^2}$, where $\psi_q = P(l = q|h = R(q))$. Since H is deterministic given L, $P(l = q|h) = 0$ for $h \neq R(q)$. We denote this generative model by $M^H = (\Theta^H, R, \Psi)$.

3.2. A theoretical result

In order to assess the transition probability $P(z^t = j|z^{t-1} = i)$, let us denote the relevant mixture models by $\Theta_j^t = \Theta^1 = \{\alpha_q, \mu_q, \Sigma_q\}_{q=1}^{K^1}$, $\Theta_i^{t-1} = \Theta^2 = \{\alpha_q, \mu_q, \Sigma_q\}_{q=K^1+1}^{K^1+K^2}$. The transition probability is formalized as the probability of transition between N-point samples from the two types

$$P(z^t = j|z^{t-1} = i) = P(S^1 \in TC_{N,\varepsilon}(\Theta^1)|S^2 \in TC_{N,\varepsilon}(\Theta^2), M^H) \quad (6)$$

where the samples S^1, S^2 are random samples from a generative model $M^H = (\Theta^H, R, \Psi)$ with unknown parameters. Specifically, we are interested in the behavior of this quantity for large N values (we used 1000 spikes per time frame) and small ε values.

The probability in Eq. (6) is proportional to the probability of generating two samples of different types from the same hidden mixture. Assessment of the latter quantities in various conditions is known as "the two sample problem" (Grosse et al., 2002; Lehmann, 1959). For our case, we show in (Bar-Hillel and Spiro, 2005, Theorem 3) the following large-sample approximation:

$$\log P(S^1 \in TC_{N,\varepsilon}(\Theta^1)|S^2 \in TC_{N,\varepsilon}(\Theta^2), M^H) \to -2N \max_R \sum_{m=1}^{K^H} \alpha_m^H \sum_{q:R(q)=m} \Psi_q D_{kl}[G(x|\mu_q, \Sigma_q)\|G(x|\mu_m^H, \Sigma_m^H)] \quad (7)$$

where

$$D_{kl}[p(x)||q(x)] = \int p(x) \log \left[\frac{p(x)}{q(x)}\right] dx \quad (8)$$

is the Kullback–Leibler divergence. $G(x|\mu, \Sigma)$ here denotes a Gaussian distribution with the parameters μ, Σ. The limit is of $N \to \infty, \varepsilon \to 0$, and $N\varepsilon^2 \to \infty$. The optimized Θ^H and Ψ appearing in Eq. (7) are given as follows:

$$\alpha_m^H = \frac{1}{2} \sum_{\{q:R(q)=m\}} \alpha_q, \qquad \psi_q = \frac{\alpha_q}{\alpha_{R(q)}^H},$$
$$\mu_m^H = \sum_{\{q:R(q)=m\}} \psi_q \mu_q,$$
$$\Sigma_m^H = \sum_{\{q:R\{q\}=m\}} \psi_q (\Sigma_q + (\mu_q - \mu_m^H)(\mu_q - \mu_m^H)^t) \quad (9)$$

The parameters of a hidden Gaussian, μ_m^H and Σ_m^H, are just the mean and covariance of the mixture $\Sigma_{\{q:R(q)=m\}} \psi_q G(x|\mu_q, \Sigma_q)$. The summation over q in Eq. (7) can hence be interpreted as the JS-divergence between the components assigned to the hidden source m, under Gaussian assumptions.

For a given parametric family, the JS divergence is a non-negative measure which can be used to test whether several samples are derived from a single distribution or from a mixture of different distributions from the given family (Lehmann, 1959). The JS divergence is computed for a mixture of n empirical distributions $P_1, \ldots, P_n$ with mixture weights $\pi_1, \ldots, \pi_n$. In the Gaussian case, denote the mean and covariance of the component distributions by $\{\mu_i, \Sigma_i\}_{i=1}^n$. The mean and covariance of the mixture distribution μ^*, Σ^* are a function of the means and covariance matrices of the components, with the formulae given in Eq. (9) for μ_m^H, Σ_m^H. The Gaussian JS divergence is given by

$$\begin{aligned} \mathrm{JS}_{\pi_1,..,\pi_n}^G(P_1, .., P_n) &= \sum_{i=1}^n \pi_i D_{kl}[G(x|\mu_i, \Sigma_i)\|G(x|\mu^*, \Sigma^*)] \\ &= H(G(x|\mu^*, \Sigma^*)) - \sum_{i=1}^n \pi_i H(G(x|\mu_i, \Sigma_i)) \\ &= \frac{1}{2}\left(\log|\Sigma^*| - \sum_{i=1}^n \pi_i \Sigma_i\right) \end{aligned} \quad (10)$$

Here $H(p(x))$ is the entropy of $p(x)$, $H(p(x)) = -\int p(x) \log p(x)\, dx$. Using the latter identity in Eq. (7), we get the following

expression for the transition log probability:

$$\log P(z^t = j|z^{t-1} = i) = -2N\max_R \sum_{m=1}^{K^H} \alpha_m^H \mathrm{JS}^G_{\{\Psi_q:R(q)=m\}}(\{G(x|\mu_q, \Sigma_q)\}) \quad (11)$$

This is referred to as the JS transition score.

3.3. Algorithms

The JS transition score (Eq. (11)) is not meaningful if we do not restrict the family of allowed hidden mixture models. Without such a restriction, we can trivially optimize the score by choosing Θ^H as the concatenation of the empirical distributions Θ^1, Θ^2 and trivial values for R and Ψ ($R(q) = q$, $\psi_q = 1$ for $q = 1, \ldots, K^1 + K^2$). For this choice we get the maximal score, which is 0. In general, a "richer" hidden model, which does not merge many Gaussians, gets a higher score. We have to impose some restriction on the hidden model to get a useful score, and we do this by posing constraints over the allowed R mappings.

Consider first the case in which a one-to-one correspondence is assumed between clusters in two consecutive frames, and hence the number of Gaussian components K is constant over all time frames. In this case, a mapping R is allowed if and only if it maps to each hidden source i a single Gaussian from mixture Θ^1 and a single Gaussian from Θ^2. Denoting the Gaussians matched to hidden i by $R_1^{-1}(i)$, $R_2^{-1}(i)$, the transition score (Eq. (11)) takes the form of $-N\max_R \sum_{i=1}^{K} S(R_1^{-1}(i), R_2^{-1}(i))$. Such an optimization of a pair-wise matching score can be seen as a search for a maximal perfect matching in a weighted bipartite graph: the nodes of the graph are the Gaussian components of Θ^1, Θ^2 and the weights of the edges are given by the scores $S(a, b)$. The global optimum of this problem can be efficiently found using the Hungarian algorithm (Kuhn, 1955) in $O(K^3)$, which is unproblematic in our case.

The one-to-one correspondence assumption is too strong for many datasets in the spike sorting application, as it ignores the phenomena of splits and merges of clusters. We wish to allow such phenomena, but nevertheless enforce strong (though not perfect) demands of correspondence between the Gaussians in two consecutive frames. In order to achieve such a balance, we place the following two constraints on the allowed partitions R:

1. Each cluster of R should contain exactly one Gaussian from Θ^1 or exactly one Gaussian from Θ^2. Hence assignment of different Gaussians from the same mixture to the same hidden source is limited only for cases of a split or a merge.
2. The label entropy of the mixture induced by R should satisfy

$$H(\alpha_1^H, \ldots, \alpha_{K^H}^H) \le \frac{1}{2}(H(\alpha_1^1, \ldots, \alpha_{K^1}^1) + H(\alpha_1^2, \ldots, \alpha_{K^2}^2)) \quad (12)$$

Intuitively, the latter constraint limits the allowed hidden mixtures to ones which are not richer than the average visible mixtures, i.e., do not have many more clusters. Note that the most detailed hidden model (the "concatenation" model) has label entropy given by the right-hand side of Eq. (12) plus 1 bit. The extra bit can be intuitively understood as the cost of using twice the number of required labels. We look for mixtures which do not pay this extra price in label information.

The optimization for this family of R does not seem to have an efficient global optimization technique, and thus we resort to a greedy procedure. Specifically, we use a bottom-up agglomerative procedure to find the partition induced by R. We start from the concatenated mixture model. This is the most detailed partition, as R is a one-to-one map and each visible Gaussian is a singleton. At each iteration of the algorithm we merge two clusters of the partition. Only merges that comply with the first constraint are considered. We look for a merge which incurs a minimal loss to the accumulated JS score, Eq. (11), and a maximal loss to the mixture label entropy. For two Gaussian clusters, $(\alpha_1, \mu_1, \Sigma_1)$ and $(\alpha_2, \mu_2, \Sigma_2)$, these two quantities are given by

$$\Delta\mathrm{JS} = -N(\alpha_1 + \alpha_2)\mathrm{JS}^G_{\pi_1,\pi_2}(G(x|\mu_1, \Sigma_1), G(x|\mu_2, \Sigma_2)),$$
$$\Delta H = -N(\alpha_1 + \alpha_2)H(\pi_1, \pi_2) \quad (13)$$

where π_1, π_2 are $\alpha_1/(\alpha_1 + \alpha_2), \alpha_2/(\alpha_1 + \alpha_2)$. We choose at each round the merge which minimizes the ratio between these two quantities. The algorithm terminates when the accumulated label entropy reduction is bigger than 1 bit or when no allowed merges exist anymore. In the second case, it may happen that the mapping R found by the algorithm violates the second constraint, Eq. (12). We nevertheless compute the score based on the R found, since this partition obeys the first constraint and usually is not far from satisfying the second.

4. Results

The algorithm was implemented in MATLAB and the code is freely available at the authors' web sites.[3] In Section 4.1 we test the algorithm in several scenarios using synthetic spike data. We experiment with varying degrees of data stationarity and noise, and compare our results to several other methods. In Section 4.2 we assess the performance of the algorithm on real electrode recordings in terms of agreement with human sorters. The agreement of the algorithm with human sorters is shown to be similar to the agreement among humans. Experiments with semi-supervised clustering are presented at Section 4.3. Guided by human solutions of every fifth or tenth frame, the algorithm adapts the rest of the clustering to the intentions of the human user.

4.1. Unsupervised clustering of synthetic data

The final test of a spike sorting system should measure its performance using real electrode recordings. However, performance in this scenario is difficult to measure since we do not have ground truth labels of the spikes. Therefore, we start with some experiments on synthetic spike trains, which allow care-

[3] http://www.cs.huji.ac.il/~aharonbh,~adams.

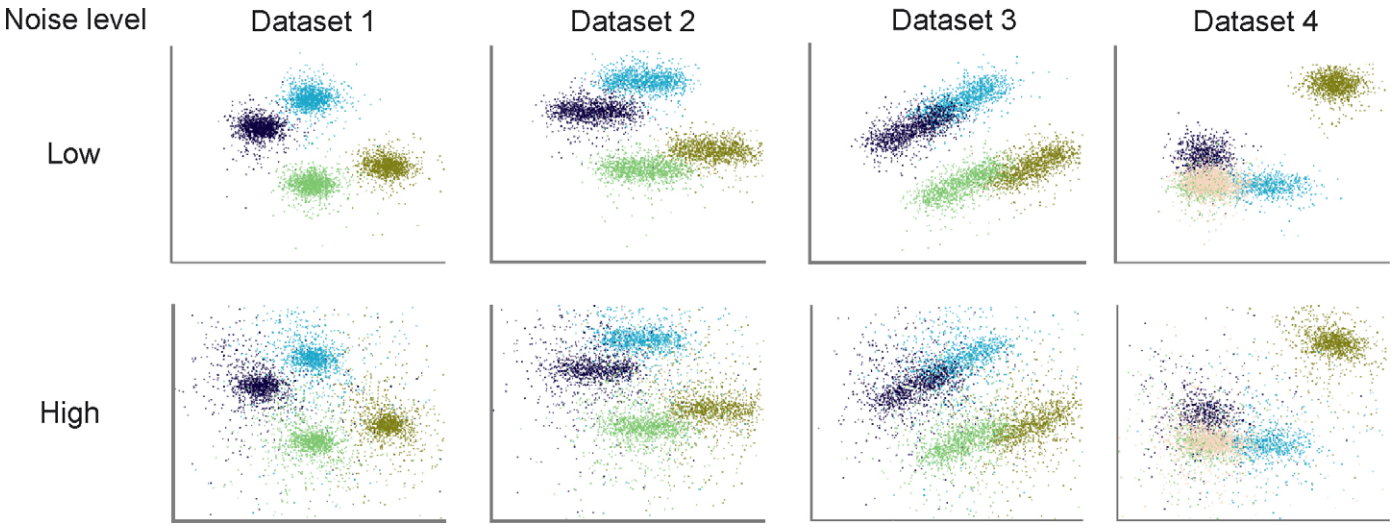


Fig. 5. Synthetic datasets used in our experiments. Top/bottom row shows data produced with a low/high noise level. Within each row, each panel shows a complete dataset in a 2D PCA plane. Datasets 2 and 3 were produced by moving the mixture from dataset 1 in different directions. Hence while clusters in these datasets are reasonably separated in any small time window, they are not well separated when considered globally. The figure is better seen in color.

ful control of the non-stationarity degree and background noise level in the data. In addition, the known ground truth labels in this setting allow us to compare our algorithm to other methods suggested in the literature.

A complete spike sorting system includes spike detection, representation and clustering. The novelty of our method is in the third stage, namely, clustering, and we use pre-existing methods for spike detection and representation (see Section 4.2 for details). We therefore concentrate here in our experiments with synthetic data on the clustering task, and synthesize a set of spike waveforms instead of a continuous signal. The spike population for a specific experiment is generated using a dynamic mixture-of-Gaussians over a 2D PCA plane. Two mixtures of Gaussians, a "start mixture" and an "end mixture" with the same number of components $K=4$ are determined. The start mixture describes the spikes distribution at time $t=0$, and the end mixture states the distribution at $t=1$. The weights of the four components in the mixture are all equal. Spike descriptions are generated from mixtures with mean and covariance parameters interpolated linearly between the start and end mixtures.

More specifically, the spikes are generated according to the following protocol. First, spike times for $N=5000$ spikes are sampled uniformly in [0, 1] and spike labels are sampled uniformly in 1, ..., K. Then spike descriptions in the 2D PCA plane are drawn. A spike generated from cluster j at time t is sampled from the mixture with mean $(1-t)\mu_j^{\text{start}} + t\mu_j^{\text{end}}$ and covariance $(1-t)\Sigma_j^{\text{start}} + t\Sigma_j^{\text{end}}$. Spikes are then projected onto 64-dimensional waveform samples using two library principal component vectors. Finally, background noise representing distant spikes is added to the spike signals. The noise is generated and controlled according to the method suggested by Quiroga et al. (2004), i.e. by placing random spikes at random times and amplitudes. Unlike Quiroga et al. (2004), who sample background spikes from a predefined library of spike shapes, we sample them from a uniform distribution over the relevant area in the 2D PCA plane. The noise level was determined from its standard deviation. Following Quiroga et al. (2004), we used standard deviations of 0.05 and 0.20 relative to the peak amplitude of the foreground spikes to simulate low and high noise levels respectively. We considered the following four scenarios:

1. A stationary dataset.
2. An "easy" non-stationary dataset, in which the clusters from scenario 1 move in the same direction but movements do not induce cluster overlaps.
3. A "difficult" non-stationary dataset, in which movements do induce overlaps.
4. A "split" scenario, in which three clusters which are identical at time $t=0$ separate into three distinct clusters at $t=1$.

For each scenario, we experimented with low and high noise levels. Fig. 5 presents the eight resulting datasets. At the split scenario (dataset 4), the "true" labeling of spikes was recomputed to reflect the inseparability of the three clusters at times close to 0. At the beginning, the Bayes error of cluster distinction[4] is high and spikes from the different clusters cannot be reliably separated. Hence the three clusters can only be regarded as a single multi-unit cluster, and we re-label the spikes from the three clusters using a single multi-unit label. At later stages, when clusters are separable enough, we keep the three original different labels. Since the clusters are Gaussians with equal covariance matrices moving away from each other at a known speed, the Bayes error in this mixture model can be computed analytically as a function of t. We chose the "split point", the point from which different labels are kept, as the point in which

[4] The Bayes error is the probability of assigning a wrong label to a spike when an ideal classifier is used.

the Bayes error drops below 0.1 (meaning that more than 90% of the spikes can be correctly labeled). Using the synthetic datasets, we compared the following sorting algorithms:

- Learning vector quantization (LVQ) algorithm, implemented by the "Spiker" package for spike sorting (Yu and Kreiman, 2000).
- An EM algorithm, applied to the whole dataset at once.
- A super paramagnetic clustering (SPC) algorithm implemented by the "Wave Clus" spike sorting package (Quiroga et al., 2004).
- Our suggested method, entitled dynamic mixture-of-Gaussians (DM), using 25 frames of 200 spikes each.

All methods were applied to exactly the same spike data, and all used the same spike representation of first two PCs. The first two methods were also given as input the correct number of clusters K, since they cannot determine it automatically (the "Spiker" package includes an option to determine K automatically, but it did not work properly and tended to choose too large K values in our experiments). SPC and DM determined K automatically. Both SPC and our method include an explicit consideration of "non-spike" events, so part of the data may be labeled as background spikes. However, since the synthetic data did not contain spurious non-spike events, we forced these algorithms to label every spike, even those which were initially labeled as background.

We quantify the agreement between two labeling suggestions of the same dataset using a combined measure of precision P and recall R scores. A clustering solution for a dataset is a discrete labels vector $L=(l_1, \ldots, l_N)$. Given two clustering solutions L^1 and L^2 with labels $A = \{a_1, \ldots, a_{K^1}\}$ and $B = \{b_1, \ldots, b_{K^2}\}$ respectively, we estimate the conditional probability matrices $P(A|B)$ and $P(B|A)$. The recall score R is based on a map $f{:}A \to B$ and is given by

$$f(a_j) = \underset{b \in B}{\operatorname{argmax}} P(b|a_j), \qquad R = \sum_{m=1}^{K^1} p(a_m) p(f(a_m)|a_m) \quad (14)$$

The precision score P is defined analogously with the roles of A and B swapped, using a second map $g{:}B \to A$. The combined agreement score $f_{1/2}$ we use is defined as

$$f_{1/2} = \frac{2PR}{P+R} \quad (15)$$

The agreement score ranges between zero (when there is no agreement at all) and one (when the agreement is perfect). Similar results were obtained when we measured agreement between solutions using the "information variation" criterion (Meila, 2003).

The quantitative results of the synthetic data experiments are summarized in Table 1. The table shows the $f_{1/2}$ scores computed between solutions produced by the various algorithms and the true spike labels. Our method performs similar to other methods when data are stationary (dataset 1), but clearly outperforms them when data are non-stationary (datasets 2–4). The advantage of DM for non-stationary scenarios is even more pronounced for high noise levels. Several examples of the clustering solutions obtained for the more difficult scenarios can be seen in Fig. 6. In these scenarios it is very hard to estimate the correct clustering based on a global view of the data in a single frame, and the three methods which assume stationarity usually fail. The analysis in small time windows allows our method to successfully track the clusters in dataset 3 and identify the split in dataset 4.

Table 1
$f_{1/2}$ agreement scores between clustering results of different algorithms and the true labels

Dataset	Noise level	LVQ	EM	SPC	DM
1	Low	0.99	0.97	0.99	0.98
2	Low	0.78	0.91	0.92	0.99
3	Low	0.75	0.84	0.75	0.99
4	Low	0.68	0.74	0.71	0.94
1	High	0.85	0.70	0.87	0.86
2	High	0.70	0.65	0.71	0.87
3	High	0.69	0.63	0.65	0.87
4	High	0.71	0.62	0.72	0.78

The synthetic datasets are illustrated in Fig. 5.

4.2. Unsupervised clustering of real data

We test our method using spike data obtained from the dorsal and ventral premotor cortices of Macaque monkeys performing a reaching and grasping task. The experimental procedures, behavioral task, and data collection were described in detail in the work of Stark et al. (2006) and are briefly summarized below. At the beginning of each trial, an object was briefly presented in one of six locations. Following a delay period, a "Go" signal prompted the monkey to reach for, grasp, and hold the target object. A recording session typically lasted 2 hours during which monkeys completed hundreds of trials. During each session 16, independently-movable glass-plated tungsten micro-electrodes (impedance 0.2–2 MΩ at 1 kHz) were inserted through the dura. Signals from these electrodes were amplified (10 K), band pass filtered (1–10,000 Hz), and sampled (25 kHz; Alpha-Map 5.4, Alpha-Omega Engineering, Nazareth Israel). Spikes were detected by computing a modified second derivative (seven samples backwards and 11 forward), accentuating "spiky" signal features (bimodal, skewed, and sharp). Segments that crossed an adaptive threshold were identified. The threshold was measured in S.D.s about the mean of the modified second derivative, and was re-computed every several minutes of recordings. The number of S.D.s to be used as a threshold were determined manually by scanning visually several minutes of recordings; typically, thresholds were set at 4–5 S.D.s. Within each segment, a potential spike's peak was defined as the time of the maximal derivative. If a sharper spike was not encountered within 1.2 ms, 64 samples (starting 10 samples before the peak) were registered as a spike waveform. Waveforms were subsequently aligned such that each started at the point of maximal fit with two library principal components (accounting, on average, for 82% and 11% of the variance (Abeles and Goldstein, 1977).

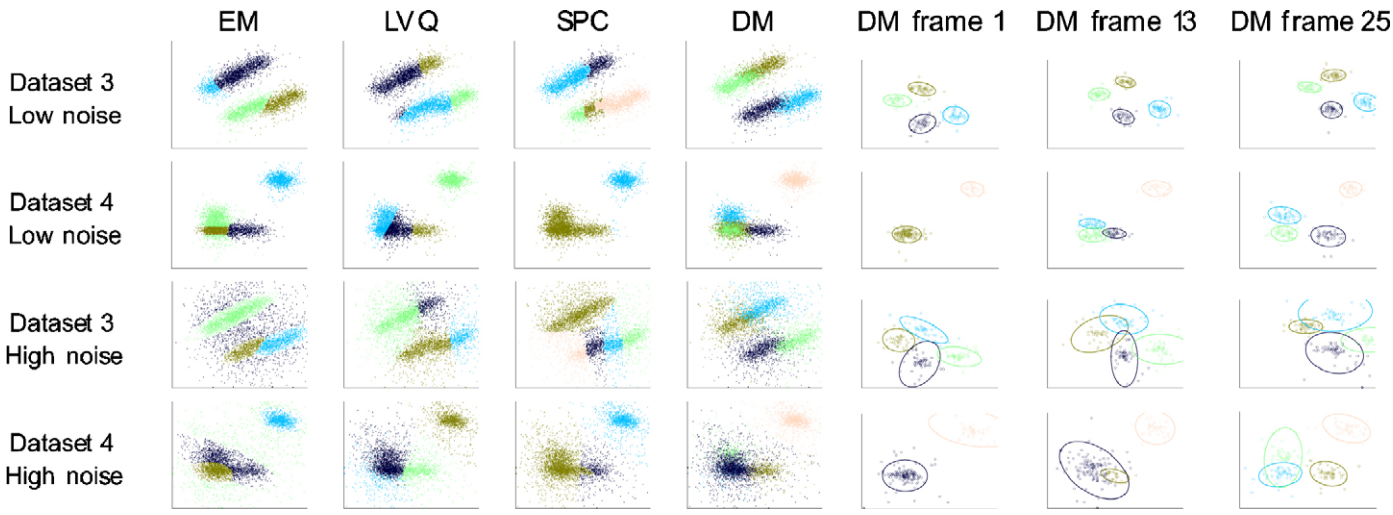


Fig. 6. Clustering results obtained for the more difficult synthetic datasets. Different datasets and noise levels are shown in different rows (see Section 4.1 for details of data generation and Fig. 5 for the labeled datasets). Clustering results for the complete datasets by four different algorithms are shown in the left four columns. For the DM algorithm, 25 frames were used. Time frames 1, 13, and 25 from the solution found by our DM algorithm are shown in the right three columns. The four clusters in dataset 3 are perfectly tracked in both noise conditions. For dataset 4, the "true labels" indicate a split of the clusters starting from frame 11. This split is well identified in the case of low background noise level, and starts from frame 12 in the solution found. In the case of a high noise level, the solution found identifies the split in two stages: the right class of the three first appears on frame 13, and then the multi-unit cluster splits again in frame 20 into the upper and middle clusters. This figure is better seen in color.

Aligned waveforms were projected onto the PCA basis to arrive at two PCs.

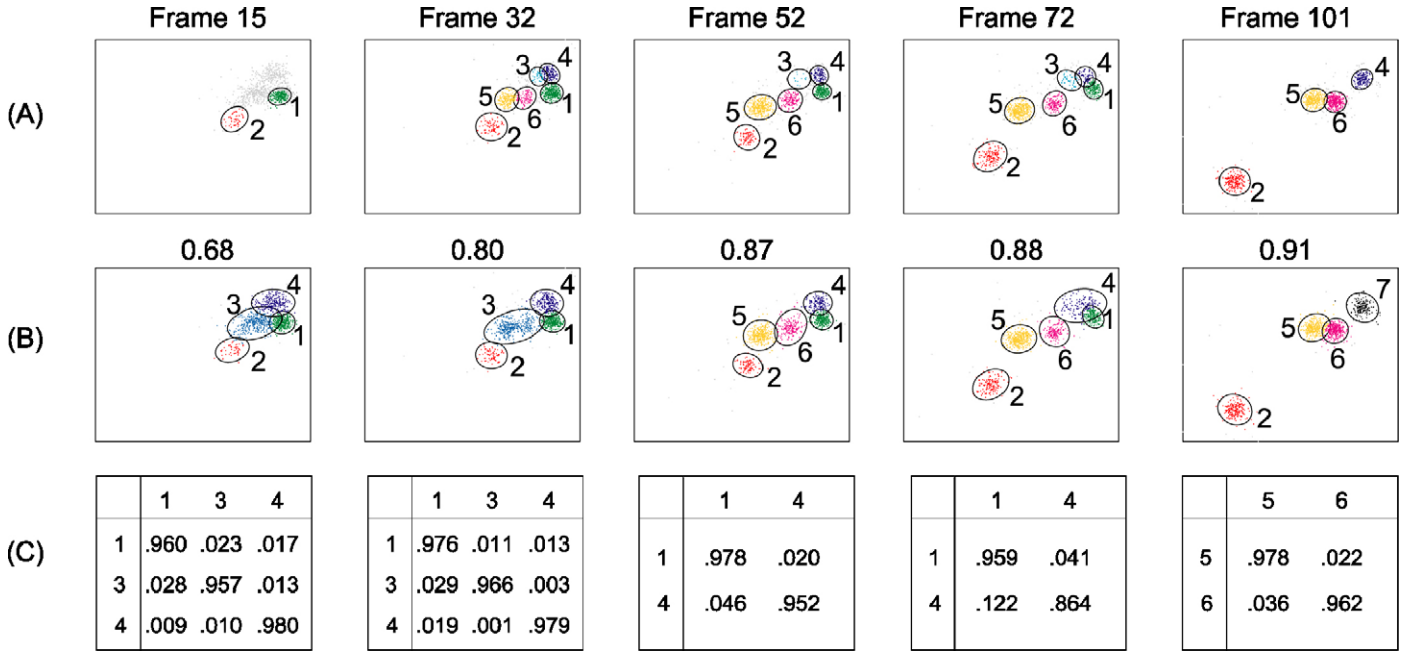
We tuned the algorithm's parameters using a training set of 46 electrode recordings. Four of these were found extremely discontinuous by a human observer and the parameters of the non-stationarity detection mechanism (Section 2.4) were tuned to identify them. The remaining 42 electrodes contained a total of 1383 time frames. The algorithm was then tested on a set of another 46 electrodes. Of these, two were automatically classified as extremely non-stationary, and statistics were computed for the remaining 44 electrodes (a total of 4544 time frames). Processing time was approximately linear with data size, averaging 12 minutes per frame (2.4 GHz Pentium IV machine with 1 GB of RAM). Spikes from train and test sets were manually clustered by a skilled user in the environment of Alpha-Sort 4.0 (Alpha-Omega Eng.). Our performance measure in this section is the agreement rate between the automated and human clustering.

Fig. 7 demonstrates the performance of the algorithm using a particularly non-stationary dataset (for more examples of clustering solutions see (Bar-Hillel et al., 2005) and our web sites). In this example, there are several differences between the algorithm and the human clustering solutions. The algorithm tended to identify less clusters per frame than the human sorter did (compare frame 32 of Fig. 7A with frame 32 of Fig. 7B) and always relabeled clusters after a split or a merge event (compare frame 101 of Fig. 7A with frame 101 of Fig. 7B). As shown in Fig. 7B, the algorithm handles non-stationarity and copes well with merges, splits, and movements of clusters.

Since we estimate a full generative model for the data, we can evaluate the probabilities of important events. One such event is misclassification of a given spike, where we are interested in the probability that the spike was not generated by its putative source. Spikes with high source ambiguity can be dropped to achieve cleaner analyses. Another measure of interest is the confusion matrix between clusters, a matrix that quantifies pair-wise cluster separation. In a confusion matrix, each term (i, j) is the probability of classifying a spike from source i as coming from source j (Fig. 7C). Thus, diagonal terms measure the probability of classifying spikes correctly, and off-diagonal terms measure the probability of erroneous classifications. For each time frame, this matrix is estimated numerically by drawing labeled samples from the mixture model of this frame. A confusion matrix for the complete data is then formed by averaging the confusion matrices of separate time frames.

We measure the label agreement of the algorithm with human clustering in two time scales: for each time frame separately, and for entire electrodes (sequences of time frames). The former gives an indication of how well the local solution matches that of the human. However, it is possible to obtain high local agreement together with low agreement scores for the entire electrode. This happens when the correspondence between clusters in two consecutive frames is not established correctly. A single correspondence failure in the middle of a sequence may completely mix two labels or split a single label into two. Thus, high agreement on the electrode as a whole requires almost perfect between-frames correspondence along the entire sequence. We compare the performance of the algorithm (*Algorithm*) to several control conditions (Fig. 8):

- Local maximum likelihood clustering (*ML*). This clustering solution is obtained by choosing at each time frame the mixture description with the highest local likelihood. Cluster correspondence between consecutive time frames is established using our standard algorithm for computation of transition



	1	3	4
1	.960	.023	.017
3	.028	.957	.013
4	.009	.010	.980

	1	3	4
1	.976	.011	.013
3	.029	.966	.003
4	.019	.001	.979

	1	4
1	.978	.020
4	.046	.952

	1	4
1	.959	.041
4	.122	.864

	5	6
5	.978	.022
6	.036	.962

Fig. 7. Manual and unsupervised clustering solutions for the non-stationary recording shown in Fig. 1. Each frame shown contains 1000 spikes, plotted here according to their first two PCs. (A) Five frames from the manual clustering solution. Frame numbers are specified above the panels and are identical to the frames shown in Fig. 1. (B) The clustering solution of the algorithm for the same frames as in (A). $f_{1/2}$ agreement scores between manual and algorithm frame clustering are given above each panel. The overall agreement score for the entire electrode is 0.72, and the mean frame-by-frame agreement is 0.84. (C) Confusion matrices of the algorithm's solution (see Section 4.2). In each confusion matrix, an entry (i, j) shows the probability that a spike from cluster i will be labeled as coming from another cluster j, $p(l_{\text{map}} = j | l = i)$; only entries for clusters with a confusion error larger than 0.01 are shown. The figure is better seen in color.

probabilities (Section 3). In this approach there is no global optimization.

- Unified clusters in the human clustering (*Unified*). Observations of the human and automated clustering solutions show that the algorithm's solution sometimes effectively unites two clusters which were considered separated in the human's solution (for example, Fig. 7, frame 32). This phenomenon usually occurs in cases in which descriptions using one or two clusters are both plausible. We hence compute the highest agreement that can be achieved by unifying two clusters in the human labeling.
- Trivial clustering (*Single label*). All spikes are assigned to one and the same cluster.
- Random clustering (*Random*). Labels are randomly assigned to spikes according to the label distribution in the manual clustering.

The average agreement score of the *Algorithm* was 0.91 for single frames and 0.774 for complete electrodes. These scores are much higher than the baseline performance of *Single label* and *Random* clustering. The *ML* solution performance for individual frames is comparable to the performance of the *Algorithm*, but gets very low scores for entire electrodes (averages: 0.91 and 0.48, respectively). This happens since mixtures are chosen independently for each time frame, and often no natural correspondence between adjacent time frames can be made.

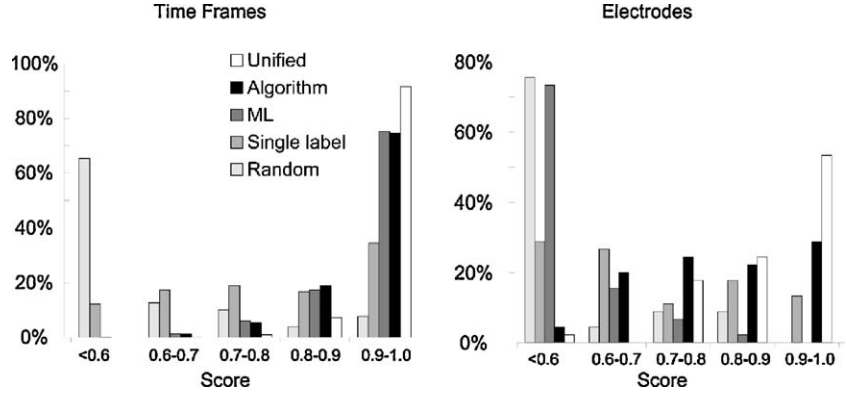


Fig. 8. Performance of the unsupervised algorithm and several control conditions assessed using the test set. The control conditions are explained in Section 4.2. The performance measures are $f_{1/2}$ agreement scores between suggested clustering solutions and human clustering. Results are shown for single time frames on the left, complete electrodes on the right.

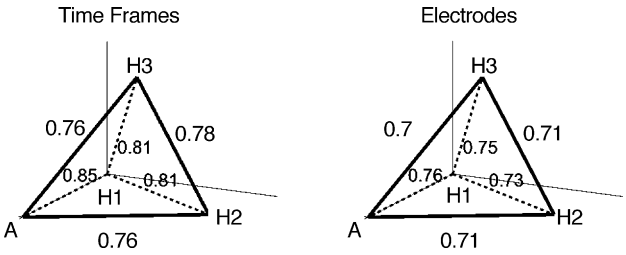


Fig. 9. Agreement scores between the automated clustering (A) and clustering solutions of three human sorters (H1, H2, and H3).

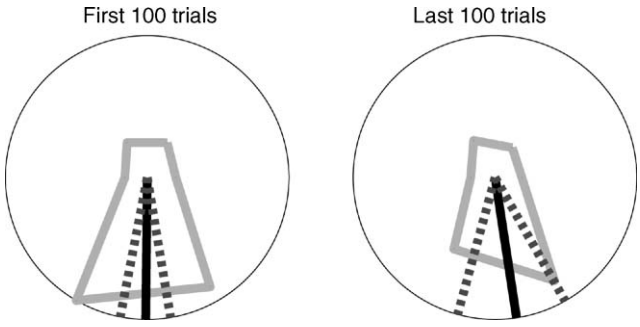


Fig. 10. Functional corroboration of clustering results. Cluster number 5 of the recording shown in Fig. 7 had clear directional tuning during the delay period. This cluster split from cluster 3 in frame 47 in the automated clustering solution and changed position in later frames. Nevertheless, its directional tuning curve (gray lines) and PD (black) were similar during the first and last 100 trials. Dashed lines are at 95% confidence limits of the PDs.

This underscores the importance of the global optimization in our method. Finally, we can see that when the human solution is modified by unification, the agreement score increases dramatically (averages of 0.92 and 0.87). This means that most of the discrepancy between human and algorithm is caused by a single additional cluster division made by the human sorter. We also obtained spike data recorded in non-cortical brain regions, the basal ganglia, and the spinal cord (sorted data courtsey of Hagai Bergman and Yifat Prut, The Hebrew University, Israel). The average agreement scores for these data were 0.87 and 0.85 for single frames and complete electrodes, respectively.

The algorithm gives reasonable, continuously evolving clustering even when there is low agreement with manual clustering. It often seems that both human and algorithm provide reasonable interpretations of an inherently ambiguous dataset. This inherent ambiguity component can be quantified by measuring the (dis)agreement between several human experts clustering the same data. To this end we obtained manual clustering solutions from three human experts (our original expert and two additional skilled sorters) for six randomly-chosen electrodes. The average agreement between the human sorters and the algorithm can be seen in Fig. 9. The average agreement of the algorithm with the human sorters (0.79 and 0.723 for single frames and electrodes, respectively) is almost identical to the average agreement among humans (0.80 and 0.73). The algorithm slightly prefers the clustering solutions of sorter 1 over the other two sorters; this is not surprising as sorter 1 is our original expert and the algorithm's parameters were tuned according to his preferences (but see Section 4.3).

In some cases, validity of the automatic clustering can be assessed by checking functional properties associated with the underlying neurons. For instance, cortical neurons often exhibit directional tuning. We thus compute the preferred direction (PD) of a unit at the beginning of a recording session and compare it with the PD at the end of the session. If the PDs are similar (Stark and Abeles, 2005), the clustering solution is corroborated. In Fig. 10 we present results of such a test for a successfully tracked cluster.

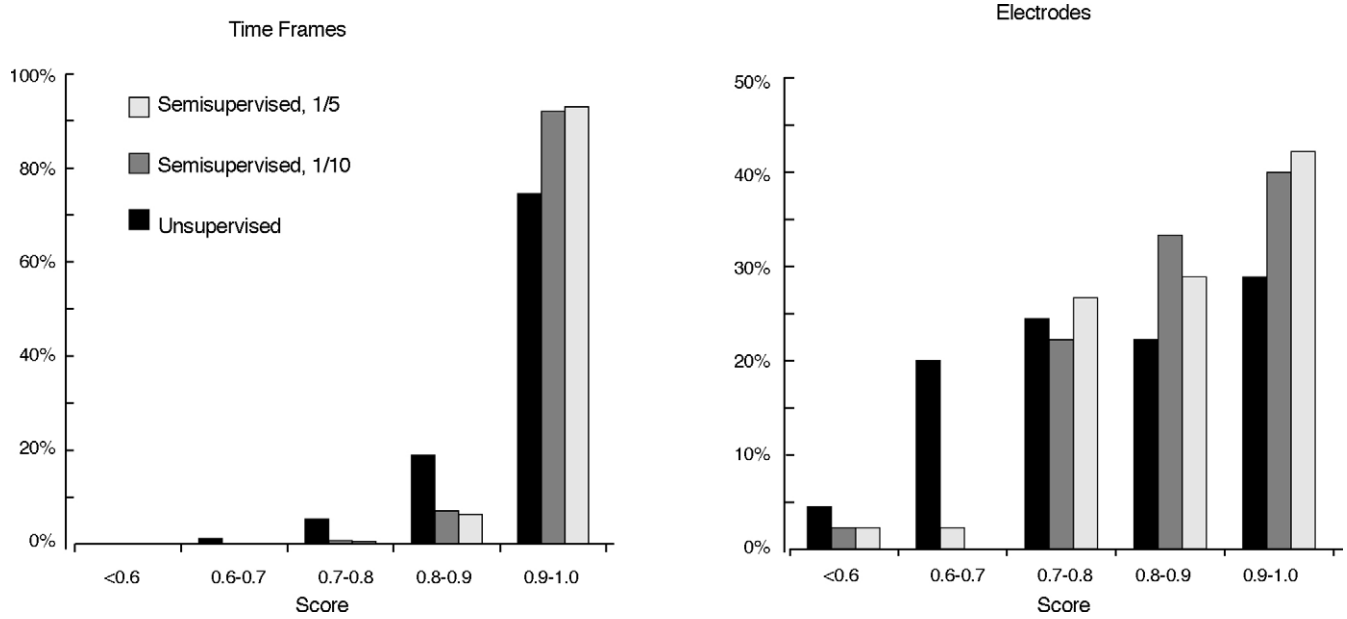


Fig. 11. Performance histograms of unsupervised and semi-supervised clustering. Results are shown for two levels of human supervision in which guidance is given for 1/5 and 1/10 of the time frames.

Table 2
Agreement scores between three human sorters and the algorithm in a semi-supervised mode

	H1 match	H2 match	H3 match
H1 supervision	**0.86**	0.70	0.71
H2 supervision	0.754	**0.785**	0.63
H3 supervision	0.778	0.664	**0.78**
No supervision	0.76	0.71	0.70

Rows in the table present results obtained with different human supervisors. 1/10 of the frames were manually clustered. The three columns present the $f_{1/2}$ agreement of the clustering found with the three human sorters. Results are given for complete electrodes, averaged over six datasets. The highest match in each row is shown in bold. The diagonal position of these high matches indicates successful adaptation of the algorithm to preferences of different sorters.

4.3. Semi-supervised clustering of real data

As mentioned in Section 2.1, human supervision is naturally incorporated into the algorithm by allowing a human user to cluster a small part of the time frames. We applied this mode of the algorithm to our test set. Two levels of human intervention were checked, in which the user provides clustering solutions for every fifth or tenth time frame. Human guidance has increased average electrode agreement of the algorithm from 0.774 to 0.837 and 0.84 using 1/10 and 1/5 of time frames, respectively; results are detailed in Fig. 11. Clearly, most of the improvement is already gained with only 1/10 of the frames clustered manually.

The main purpose of the semi-supervised clustering mode is to easily adapt the automatic algorithm to the clustering preferences of a specific user. We checked whether in this mode the algorithm can adapt to the preferences of different users, using the six datasets for which we have three different manual clustering solutions. The results with manual clustering of 1/10 of the data are given in Table 2. Supervision increased the agreement of the algorithm with the supervisor by more than 10%. It allowed sorters 2 and 3 to override the initial tendency of the algorithm toward sorter 1, the sorter whose labels were used in tuning the parameters of the algorithm.

5. Discussion

We presented a fully automated spike sorting method that copes well with non-stationary clusters, including merges and splits. The main motivation for the development of the method is to save human labor. Empirical tests with synthetic spike data showed that the method has a clear advantage over existing methods in handling non-stationary cases. Experiments with real data showed that the method practically reaches human performance level, judged by measuring the agreement between clustering solutions. These results were achieved using an algorithm which basically mimics the behavior of human experts. There are several main factors that explain the success of the method in achieving human-like performance. First is the reasonable statistical model of a chain of Gaussian mixtures which closely follows human intuition. The global optimization, which introduces past and future information into the clustering decision at each specific frame, is the second decisive factor. Last, a module of rule-based, classic AI system mimics human treatment of exceptional cases.

Our chain of mixtures-of-Gaussians model relies on several statistical assumptions. In particular, we assume large samples and approximate local stationarity. However, the added rule-based mechanisms (Section 2.4) allow the method to cope with failure of these assumptions. Problems of small sample size are treated by finding "sparse" clusters in a large time window, and failure of local stationarity is identified and reported. Therefore the complete algorithm makes very few explicit assumptions. While the main statistical module is almost parameter free, the added rule-based mechanisms require quite a few parameters. However, no parameter value is critical, and a wide range of values can be used.

The algorithm does not use all information contained in the data and hence is clearly not optimal. First, working with two-dimensional spike representations implies that important information present in the original waveform is dropped (approximately 7% of the spike waveform variance, Abeles and Goldstein, 1977). Using low-dimensional representation we cannot adequately account for the problem of overlapping spikes (Lewicki, 1994; Zhang et al., 2004). However, this problem can be addressed independently, before spike sorting (Hulata et al., 2002). Our method can be naturally augmented to handle spike representations in higher dimensions. Specifically, using three PCs instead of two can account for an additional 3% of the spike variance with minor changes in the existing system. Another possibility is to try and use wavelets features, which may be superior to PCA coefficients at certain scenarios (Hulata et al., 2002; Quiroga et al., 2004). We did not follow these paths because our main evaluation criterion was the agreement with human experts, and they rely upon visual representations in two PCA dimensions. A second source of information we disregarded was the detailed temporal specifics, such as the ISIs. Using refractory period considerations, constraints of the form "spike 1 cannot be from the same cluster as spike 2" can be extracted. Such constraints were used to guide clustering decisions (Fee et al., 1996; Snider and Bonds, 1998) and can be incorporated into the EM algorithm to guide the search for good local solutions (Shental et al., 2003). We neglected this information here since decisive constraints were rare in our data, occurring for about 1% of the spikes.

The agreement between clustering solutions of different human experts was found to be rather low in our experiments (Fig. 9). These results are consistent with other studies measuring the accuracy of human clustering when the "ground truth" is known. In the work of Harris et al. (2000), spikes were recorded using intra-cellular and extra-cellular electrodes simultaneously, and intra-cellular measurements served as the ground truth. Human errors reported were up to 30%. Similar human error rates were reported for carefully designed synthetic data (Wood et al., 2004a). In view of these findings, one may question whether mimicking a human expert should be the goal of an automated spike sorting system. Human sorters use high level considerations, and deal very well with exceptional cases: it is

unlikely that an automated system will outperform humans in these respects. However, three lines of evidence suggest that the overall performance of a well-designed, automated system may be *superior* to human performance. First, while human clustering is subjective and may vary with time and person, the automated method clearly does not suffer from such internal variance. Second, it is pointed in the work of Harris et al. (2000) that a major factor for human errors is human inability to efficiently visualize and make decisions regarding high dimensional data. As noted above, high dimensional data can be easily handled by our method. Finally, automated sorting can be done in a Bayesian framework using statistically sound models. This leads to accurate determination of cluster borders, which may be better than intuitive human choice. In addition, such a framework yields natural confidence measures for the spike labels, as the probability of label confusion can be inferred from the model (Fig. 7C).

Based on the considerations mentioned above, it is reasonable to assume that extending the method proposed here to spike representations in higher-than-two dimensions will yield a better-than-human performance. Specifically, this may be the case for tetrode data, which are much harder to visualize. The main obstacle for going beyond human performance (and the reason we did not try it in this work) is the lack of a large corpus of data with reliable ground truth labels. Such a corpus is required both for the parameter tuning of the algorithm and for evaluation of its performance. One way to circumvent this problem would be to evaluate spike sorting indirectly using decoding performance measures, as done in Wood et al. (2004b). In this work a Gaussian mixture approach to spike sorting was used and performance was evaluated by prediction of hand kinematics. The predictions based on the algorithm were shown to be better than predictions obtained using manually sorted data. A problem with this approach is that since sorting and decoding are jointly evaluated, it is hard to learn a stable sorting strategy which is independent of specific task and decoding algorithm. Another possible approach is to obtain a large corpus of labeled data using intricate experimental procedures (Harris et al., 2000).

In sum, our contribution here is a robust method for spike sorting which achieves close-to-human performance level. The method handles non-stationary events, background noise, and sparse clusters. The algorithm presented can be adjusted to specific clustering preferences in a semi-supervised mode, saving considerable time and manual labor. Finally, the use of a generative statistical model enables estimation of spike misclassification probabilities and clustering confusion matrices, valuable tools for further data analyses.

Acknowledgements

We are grateful to Moshe Abeles and Itay Asher for their help in spike sorting and for insightful comments. We thank Hila Zadka and Yifat Prut for providing sorted spike data. We thank Daphna Weinshall for valuable discussions and for her continuing support. Aharon Bar-Hillel was supported by a Horowitz foundation grant.

References

Abeles M, Goldstein MH. Multispike train analysis. P IEEE 1977;65:762–73.
Bar-Hillel A, Spiro A. Non-stationary data clustering using type-based transition probabilities. Hebrew University of Jerusalem (http://www.cs.huji.ac.il/~aharonbh/papers/TechRep05.pdf), 2005.
Bar-Hillel A, Spiro A, Stark E. Spike sorting: Bayesian clustering of non-stationary data. In: Proceedings of the 18th International Conference on Neural Information Processing Systems, vol. 17; 2004. p. 105–12.
Cover TM, Thomas JA. Elements of information theory. New York: Wiley; 1991. pp. 279–286.
Csiszar I. The method of types. IEEE T Inform Theory 1998;44:2505–23.
Dempster AP, Laird NM, Rubin DB. Maximum likelihood from incomplete data via the EM algorithm. J Roy Stat Soc B 1977;39:1–38.
Emondi AA, Rebrik AP, Kurgansky AV, Miller KD. Tracking neurons recorded from tetrodes across time. J Neurosci Meth 2004;135:95–105.
Fee M, Mitra P, Kleinfeld D. Automatic sorting of multiple unit neuronal signals in the presence of anisotropic and non-Gaussian variability. J Neurosci Meth 1996;69:175–88.
Grosse I, Bernaola JP, Carpena P, Roman RR, Oliver J, Eugene SH. Analysis of symbolic sequences using the Jensen–Shannon divergence. Phys Rev E 2002;65:041905.
Harris KD, Henza DA, Csicsvari J, Hirase H, Buzsaki G. Accuracy of tetrode spike separation as determined by simultaneous intracellular and extracellular measurements. J Neurophysiol 2000;84:401–14.
Hulata E, Segev R, Ben-Jacob E. A method for spike sorting and detection based on wavelet packets and Shannon's mutual information. J Neurosci Meth 2002;117:1–12.
Kuhn HW. The Hungarian method for the assignment problem. Nav Res Log 1955;2:83–97.
Lehmann EL. Testing statistical hypotheses. New York: Wiley; 1959.
Lewicki MS. Bayesian modeling and classification of neural signals. Neural Comput 1994;6:1005–30.
Lewicki MS. A review of methods for spike sorting: the detection and classification of neural action potentials. Network 1998;9:R53–78.
Mackay D. Information theory, inference, and learning algorithms. Cambridge: Cambridge University Press; 2003. pp. 324–333.
Meila M. Comparing clusterings by the variation of information. In: Proceedings of the 16th Annual Conference on Learning Theory; 2003. p. 173–87.
Quiroga RQ, Nadasdy Z, Ben-Shaul Y. Unsupervised spike detection and sorting with wavelets and superparamagnetic clustering. Neural Comput 2004;16:1661–87.
Shental N, Bar-Hillel A, Hertz T, Weinshall D. Computing Gaussian mixture models with EM using side information. In: The continuum from labeled to unlabeled data in machine learning and data mining, a workshop in the 20th International Conference on Machine Learning; 2003.
Shoham S, Fellows MR, Normann RA. Robust automatic spike sorting using mixtures of multivariate *t*-distributions. J Neurosci Meth 2003;127:111–22.
Snider RK, Bonds AB. Classification of non-stationary neural signals. J Neurosci Meth 1998;84:155–66.
Stark E, Abeles M. Applying resampling methods to neurophysiological data. J Neurosci Meth 2005;145:133–44.
Stark E, Asher I, Drori R, Abeles M. Encoding of reach and grasp by single neurons in the premotor cortex is independent of recording site, 2006, submitted for publication.
Wood FD, Black MJ, Vargas-Irwin C, Fellows M, Donoghue JP. On the variability of manual spike sorting. IEEE T Bio-Med Eng 2004a;51:912–8.
Wood FD, Fellows M, Donoghue JP, Black MJ. Automatic spike sorting for neural decoding. P IEEE Eng Med Biol Soc September 2004b:4009–12.
Yu A, Kreiman G. Spiker—a spike sorting package. Massachusetts Institute of Technology, (http://ramonycajal.mit.edu/kreiman/academia/spike_sorting.html), 2000.
Zhang PM, Wu JY, Zhou Y, Liang PJ, Yuan JQ. Spike sorting based on automatic template reconstruction with a partial solution to the overlapping problem. J Neurosci Meth 2004;135:55–65.