Chapter 5 Spike Sorting

5.1 Introduction

In this chapter we take up the first and most extensive of our neural data-analytic applications of latent variable methods. Spike sorting allows scientists and technologists to efficiently and reliably monitor the signals emitted simultaneously by many different nerve cells within intact brains. To neuroscientists, interested in how the brain carries out it complex functions, such multi-neuron data is essential input to improved understanding. In addition, the ability to collect signals from large numbers of specific neurons brings biomedical engineers closer to the dream of prosthetic devices driven directly by neural output.

5.1.1 Extracellular recording: the source and nature of the signal

The action potential

Most neurons communicate with each other by means of short, local perturbations in the electrical potential across the cell membrane, called **action potentials**. The discovery of the mechanism that gives rise to the action potential was one of the seminal breakthroughs of early neurophysiology (Hodgkin and Huxley 1952), and the account made at that time of action potentials in the squid giant axon has proven to apply quite broadly. For the purposes of this discussion, we will not need a detailed account of the action potential. However, a qualitative understanding of some points will be important.

Protein complexes embedded in the membranes of neurons pump specific ions into or out of the cytoplasm so as to establish strong concentration gradients across the membrane. The membrane possesses a baseline permeability to some of these ions, and so the system equilibrates with an electrical potential opposing the chemical potential established by the ion pumps. This electrical potential, around -70 mV for most cells (the convention is that membrane potentials are measured inside the cell, with reference to the extracellular medium), is known as the **resting potential**. Cells at rest are said to be **polarized**. Two ions are important to the action potential. Sodium ions (Na^+) are concentrated outside the cell at rest, while potassium ions (K^+) are concentrated inside.

Besides the ion pumps, the membrane contains other proteins that serve as temporary **channels** to specific ions. These channel proteins have two or more metastable conformations. In one of these, the **open** conformation, the channel allows specific ions to pass through it. Thus, as the number of channels in the open state varies, the permeability of the membrane to specific ions changes. Two

types of channel, one permeable to Na^+ and the other to K^+, form the basic machinery of the action potential. Both channels are voltage-sensitive, that is, the probability of finding them in the open state depends on the electrical potential across the membrane. In particular, they are both more likely to open as the potential inside the cell increases.

The action potential is initiated when a patch of membrane becomes slightly depolarized. As the interior voltage increases, the voltage-sensitive sodium channels are faster to open that the potassium ones. Na^+ ions are driven into the cell through these open channels, further raising the interior potential and establishing a rapid positive-feedback loop. This feedback loop is terminated in two ways. First, once in the open state, the sodium channels begin to transition to a third, **inactivated** conformation. Here again the channel is impermeable to ions, but this configuration is different from the original, closed, one. In particular, the probability of transition back into the open state, while the membrane potential remains high, is now extremely low. The return transition, called **de-inactivation**, happens only at potentials near or below rest, when the protein switches directly to the closed state. Second, the potassium channels also open in response to the increased cellular potential. The diffusion gradient for K^+ is opposite to that for Na^+, and so K^+ ions leave the cell, restoring its polarization. In fact, the membrane potential falls below the resting level. As it falls, the potassium channels close (they have no inactivated state). Eventually, all of the voltage-sensitive channels are either inactivated or closed, returning the membrane to its baseline permeability and the resting potential.

The voltage-sensitive sodium channels are most highly concentrated on the cell body at the point where the axon emerges (the axon hillock). This is the first piece of cell membrane to undergo an action potential, usually initiated by the passive propagation of depolarizations caused by membrane channels in the dendrite that open due to synaptic input. This action potential depolarizes a nearby piece of membrane on the axon, thus launching it into an action potential too, which, in turn, depolarizes a further piece and so on. Thus, once initiated at the hillock, the action potential travels down the axon, eventually triggering the release of a neurotransmitter onto another cell.

As the membrane comes out of the action potential, a number of potassium channels are still open and many sodium channels remain inactivated. Thus, for a short period of time called the **absolute refractory period** it is impossible to induce a second action potential in the cell. Even after the potassium channels have all closed and enough sodium channels have de-inactivated to allow another action potential to begin, the threshold perturbation needed to seed the action potential will be higher than normal. This period is called the **relative refractory period**. Eventually the inactivation of the sodium channels drops to an equilibrium level and the cell returns to the rest state.

In many cases a cell will fire a group of action potentials spaced by little more than the absolute refractory period. Such a group is called a **burst** or, sometimes, a **complex spike**. In general,

such bursts are not driven entirely by synaptic input, but rather by the biophysics of the neuronal membrane. For example, extremely long time-constant voltage-sensitive calcium channels are found in some neurons. The first action potential in a burst causes some number of these to open, but they neither close nor inactivate rapidly. Ca^{++}, which is concentrated outside the cell by the ion pumps, flows in through these open channels. As a result, as soon as the first action potential is over and the potassium channels closed, the depolarizing calcium current can launch the next action potential. The cell is still in its relative refractory period, however, so many sodium channels are still inactivated. As a result, the currents that flow in this and subsequent action potentials may not be quite as strong as in the initial one.

In many, if not most, neurons, voltage-sensitive channels are to be found all over the cell body and dendritic surface. Recent work in pyramidal neurons has shown that the action potential propagates not only down the axon, but also from the axon hillock back into the dendrite (Stuart and Sakmann 1994; Stuart *et al.* 1997; Buzsaki and Kandel 1998). Further, the degree of penetration varies with the recent activity of the cell (Spruston *et al.* 1995; Svoboda *et al.* 1997). The later action potentials in a burst penetrate the dendrite to a much lesser degree than the first.

Extracellular recording

The mechanism of the action potential, as well as many other important neuronal phenomena, have been understood through measurements taken using an intracellular electrode, that is, one which penetrates the cell. Unfortunately it is difficult to record with such an electrode in an intact animal and all but impossible in many awake ones. Fortunately, if all that is needed is the timing of action potentials in the cells, it is possible to acquire this information with an extracellular electrode. The most common such electrode is a fine metal wire, insulated everywhere but at the tip, which is tapered to an extremely fine point of only a few microns diameter. The uninsulated tip acquires a layer of ions at its surface which form the second plate of an extremely thin capacitor. The resistive coupling of the electrode to the surrounding medium is generally weak; resistances in the hundreds of $M\Omega$ are not uncommon. However, the capacitive coupling is much stronger, with 1kHz impedances in the hundreds or thousands of $k\Omega$.

The electrical currents associated with the flow of ions through the membrane are transient. If the electrode tip is near the membrane surface during an action potential, these currents couple to the electrode, resulting in a transient change in the potential of the electrode measured relative to any stable external point. Thus, if we were to make a trace of the electrode potential over time, we would see **spikes**[1] in the trace corresponding to the action potentials in the cell near the tip. The

[1] In this chapter, "spikes" occur in the electrode voltage trace, while "action potentials" occur on the cell membrane. This sharp distinction is not entirely conventional, but it is useful, allowing us to speak, for example, of the "changing amplitude of a spike" without any implications about the maximal currents that flow across the cell membrane. The time of occurrence of the spike and action potential will be taken to be the same.

relationship between the intracellular trace of the action potential and the extracellularly recorded spike is complex. First, the extracellular probe records a integral current from many patches of membrane that may be in many different stages of the propagating action potential. Second, the tip geometry filters the measured spike; for an electrode with a smooth surface this filter is dominated by a single-pole high-pass component, but for porous electrode tips (plated with platinum black, for example) it is more complicated (Robinson 1968).

Many cells' membranes might lie close to the electrode tip so that spikes from many cells are recorded. Historically, the experimenter has manoeuvered the electrode so that the tip lies very close to one cell, and thus the spikes from this cell are far larger in amplitude than the spikes from other cells. A simple hardware device can then be used to record the times of these large spikes, and thus of the action potentials in a single cell. Even if the spike shape associated with the neuron varies, its amplitude remains greater than that of any other cell's spikes. This process is referred to as single-cell isolation. It is time-consuming and, in an awake animal, temporary. Movement of the tissue relative to the electrode eventually causes the experimenter to "lose" the cell.

Multineuron recording

One can only learn so much about the brain by monitoring one neuron at a time. As a result, there has been a great deal of recent interest in multineuron recording[2].

There is some reason to believe, based on the biophysics of neurons (the literature is extremely large, but see, for example, Softky and Koch 1993) as well as some direct experimental evidence (again a list of citations could be very long, so we choose a recent example: Usrey *et al.* 1998), that action potentials that occur simultaneously in a pair of neurons with a shared synaptic target are far more effective at causing the target to fire than are two non-coincident action potentials. It is possible, then, that coincident firing plays a significant role in the transmission of information within the nervous system. A number of experimenters have argued that indeed more, or different, information is available if the precise timing of action potentials across multiple cells is taken into account (e.g., Gray and Singer 1989). Furthermore, even if the exact relationship of firing times between cells is not functionally significant, this relationship can provide valuable (though indirect) clues to the micro-circuitry of the system (e.g., Alonso and Martinez 1998; Abeles *et al.* 1993).

It is possible to collect multineuron data by introducing many separate electrodes into the brain and isolating a single neuron with each one. Indeed many of the studies cited above were carried out in this way. This approach is, however, difficult to execute and difficult to scale. There are two approaches possible to obtaining many isolations. One can insert many individually positionable

[2]We shall take "multineuron recording" to mean that separate (or separated) spike trains from multiple cells are available. This situation is sometimes called "multiple simultaneous single-neuron recording" to distinguish it from the earlier use of the term "multineuron recording" which was applied to a single spike train representing all the action potentials in an unknown number of cells near the electrode tip. This earlier usage seems to be fading as technology advances, and the term "multineuron" is less cumbersome than "multiple simultaneous single-neuron".

electrodes and manoeuver each to isolate a cell, or one can insert a larger number of fixed electrodes and simply record from those that happen to provide a decent isolation. The former approach requires considerable time from the experimenter. Furthermore, since, at least in awake animals, isolations generally last for only a short time, as the experimenter isolates cells on more and more electrodes he risks losing the cells isolated at the outset. The latter of the two approaches will often lead to a more stable recording than can be obtained with manoeuverable electrodes, in part because the probes can be allowed to settle within the tissue over a long time. However, the yield of electrodes with single-cell spike trains can be extremely low.

5.1.2 Spike sorting

Spike sorting provides an alternative to physical isolation for multineuron recording. In this approach, the electrode is placed in the neuropil, with no effort being made to isolate a single cell. Instead, the spikes due to many cells are recorded and a data-analytic effort is made to sort them into groups according to their waveforms. Each such group is presumed to represent a single cell.

The attractions to this approach are clear. If repositionable electrodes are used, far less manoeuvering is needed in order to obtain clear spike information. If fixed electrodes are used, the yield of recordable cells from a given array is much increased. Beyond such issues of experimental efficiency, spike sorting approaches can provide data that is extremely difficult to obtain using one-cell-one-electrode approaches. All the cells detected on a single electrode lie within some few tens of microns of the tip, and thus of each other. Such cells are more likely to be functionally and anatomically related than well-separated neurons chosen at random.

Multiple-tip electrodes

Spike sorting can be made easier by use of a multi-tip electrode such as a stereotrode[3] (McNaughton *et al.* 1983) or tetrode (Recce and O'Keefe 1989). This is really a group of electrodes whose tips lie sufficiently close together that an action potential in a single cell generates a spike on more than one of the electrodes. Each electrode will have a different spatial relationship to the source cell, and so experience a slightly different spike waveform. Put together, these "multiple views" of the same action potential provide more information on which to base the sorting of the spikes.

An analogy may be drawn to stereophonic sound recording. Two instruments with similar timbre cannot be distinguished in a monophonic recording. With two microphones, the added spatial information allows us to hear the two different sources. This analogy can only be taken so far, however. In the stereophonic recording the scale of the separation between sources and microphones is very much greater than the scale of the sources and microphones themselves. This is not the

[3] Unfortunately, the term "stereotrode" has come to mean a two-wire electrode. We shall continue in this usage, even though a tetrode, with its four wires, is as much a stereotrode as its two-wire predecessor.

case in the neurophysiological recording. The tip size, the distance from the membrane and the segment of membrane that contributes to each recorded spike are all on the order of 10 microns. As a result, some of the simple sorting strategies suggested by the recorded music analogy are not actually workable.

5.2 Data Collection

The algorithms that appear in this chapter are expected to be of general applicability. They have been developed, however, with reference to data taken in two preparations: parietal cortex of macaque monkey[4] and locust lobula[5]. The methods of data collection are described here.

5.2.1 Monkey

Data have been collected from two adult rhesus monkeys (*Macaca mulatta*). A stainless steel head post, dental acrylic head cap, scleral search coil, and stainless steel recording chamber were surgically implanted in each monkey using standard techniques (Mountcastle *et al.* 1975; Judge *et al.* 1980). During recording, the monkeys sat in a primate chair (custom); the implanted head posts were secured to arms attached to the chairs, thereby immobilizing the animals' heads. Eye-positions were monitored in two dimensions by recording the level of *emf* induced in the scleral coil by two external magnetic fields that oscillated at non-reducible frequencies (Fuchs and Robinson 1966).

The recording chambers in each monkey were set over a craniotomy opened over the posterior parietal cortex. All electrodes were inserted in this area; in most cases they penetrated to the lateral intra-parietal area (LIP). During recording, the animals were awake and performing a "memory-saccade" task in which they remembered the location of a flash of light and then looked towards it on a cue. The details of the task will not be relevant to the present discussion.

In all cases a single tetrode was used for recording (Pezaris *et al.* 1997). The tetrodes were prepared from 13μm-diameter tungsten wire (California Fine Wire), insulated along its entire length. Four strands of wire were twisted together at approximately 1 turn/mm and heated so that the insulation fused over a length of some 10cm. One end of the fused region was cut with sharp scissors so that the tungsten conductor was exposed in all four strands. The impedance of the each conductor interface to physiological saline was measured to be between 0.4 and 0.7 MΩat 1kHz. At the other end the four strands remained separated and were individually stripped of their insulation with a chemical stripper and bonded with conductive paint to electrical connectors.

The tetrode was inserted into a construction of nested metal cannulae which provided mechanical support. The tip of the narrowest, innermost, cannula was sharpened and inserted through the dura

[4] Data collected in collaboration with J. S. Pezaris in Dr. R. A. Andersen's laboratory.
[5] Data collected in collaboration with M. Wehr and J. S. Pezaris in Dr G. Laurent's laboratory.

mater, with minimal penetration of the underlying neural tissue. The tetrode could then be advanced from within this cannula into the brain by a hydraulic microdrive (Frederick Haer Company). A series of tests in another animal revealed that the tetrodes tend to travel straight once inserted into the brain.

The electrical connector at the end of the tetrode was inserted into an amplifier head-stage (custom) with 100x gain. The animal, electrode and head-stage amplifier were all placed within an electromagnetically shielded room. Amplification was in differential mode, with the cannula assembly serving as the reference electrode. Four coaxial cables fed the signals from the head-stage amplifier to the main amplifier (custom) with adjustable gain. Besides enhancing it, the amplifiers also reversed the polarity of the signal. This resulted in the peak amplitude of each spike appearing positive, rather than negative as is the case at the electrode tip. We will maintain this convention throughout the chapter.

The amplified signals were filtered to prevent aliasing and digitized. The digitization rate at the A/D converters (Tucker Davis Technologies AD-2) varied between 12.8 and 20 kHz. The 9-pole Bessel low-pass anti-aliasing filters (Tucker Davis Technologies FT5-4) had corner frequencies of either 6.4 or 10kHz. The data were recorded to digital media and all subsequent operations performed off-line, although sometimes under simulated on-line conditions.

5.2.2 Locust

A difficulty common to almost all data sets used for the development of spike sorting techniques is ignorance of the ground truth. There is no independent way in which to establish the number of distinct cells whose spikes are present in the recording, nor to know which cell fired when. These data, collected from the lobula of the locust, were collected in an attempt to remedy at least one of these concerns. Recordings were carried out with a single tetrode as well as two sharp pipette, intracellular, electrodes. The intracellular electrodes provided incontrovertible information about the firing of up to two cells in the region. Often, one or both of these cells would invoke sizable spikes on the tetrode.

Experiments were carried out *in vivo* on adult female locusts (*Schistocerca americana*). Animals were restrained dorsal side up, the head was immobilized with beeswax, and a watertight beeswax cup was built around the head for saline superfusion. A window was opened in the cuticle of the head capsule between the eyes, and air sacs on the anterior surface of the brain carefully removed. For stability, the oesophagus was sectioned anterior to the brain, and the gut removed through a subsequently ligatured distal abdominal section. The brain was treated with protease (Sigma type, XIV), gently desheathed, and supported with a small metal platform. The head capsule was continuously superfused with oxygenated room-temperature physiological saline (in mM: 140 NaCl, 5 KCl, 5 $CaCl_2$, 4 $NaHCO_3$, 1 $MgCl_2$, 6.3 HEPES, pH 7.0).

Intracellular recordings were made using conventional sharp glass microelectrodes pulled with a horizontal puller (Sutter P-87), filled with 0.5 M KAc, for resistances of 100–300 MΩ. Intracellular recordings were done in bridge mode using an Axoclamp 2A amplifier (Axon Instruments) from the third optic lobe (lobula). Data were collected from 28 single neuron and 6 paired intracellular recordings, all with simultaneous tetrode recordings, from 7 animals. The tetrode was prepared as described above.

All signals were amplified, low-pass filtered at 10 kHz (8-pole analogue Bessel with gain, Brown-Lee Precision), digitized at 50 kHz with 16-bit resolution (Tucker Davis Technologies), and written to compact disc.

5.3 A Generative Model Schema for Extracellular Recording

The cornerstone of our approach to spike sorting will be the identification of an adequate generative model for the observed extracellular recording data. The model has to be powerful enough to account for most of the variability observed in the data, while being simple enough to allow tractable and robust inference. In fact, we will identify not one model, but a **model schema**, that is, a group of models of similar structure. The choice of a particular model from within this schema will be made on a case-by-case basis, using data-driven model selection procedures.

The recorded signal is dominated by the firing of nearby cells; in general the thermal noise in the electrode and noise in the amplification system can be neglected relative to the neural signal. For a 0.5 MΩelectrode at 300K (treated as a purely capacitive impedance) the root-mean-square amplitude of the thermal noise integrated over a 10kHz bandwidth is on the order of 5μV. As we will see (for example, see figure 5.2), this is generally smaller than the recorded amplitudes of neural signals.

We divide the cells into two groups — **foreground** and **background** — of which the second is much the larger. The division is somewhat arbitrary. Roughly, the foreground cells are those whose influence on the recorded signal is large enough that we expect to be able to recognize and sort spikes that arise from them, while the background cells are so distant that their spikes merge into an indistinguishable baseline. In practice, there will be cells whose spikes are occasionally, but not always, distinguishable. We treat these as foreground cells in the model, detecting those spikes that rise out of the background, but neglect the data thus obtained as unreliable.

Thus, we think of the recorded signal as the superposition of spikes from the foreground cells and a single, continuous background **noise process**, which is itself the superposition of all the spikes from the background cells, and other noise sources. Provided that the currents do not total to a sum that is beyond the ohmic limit of the intracellular medium, we expect each of these superpositions to be linear. Measurements made in the locust lobula show that at least in that preparation they

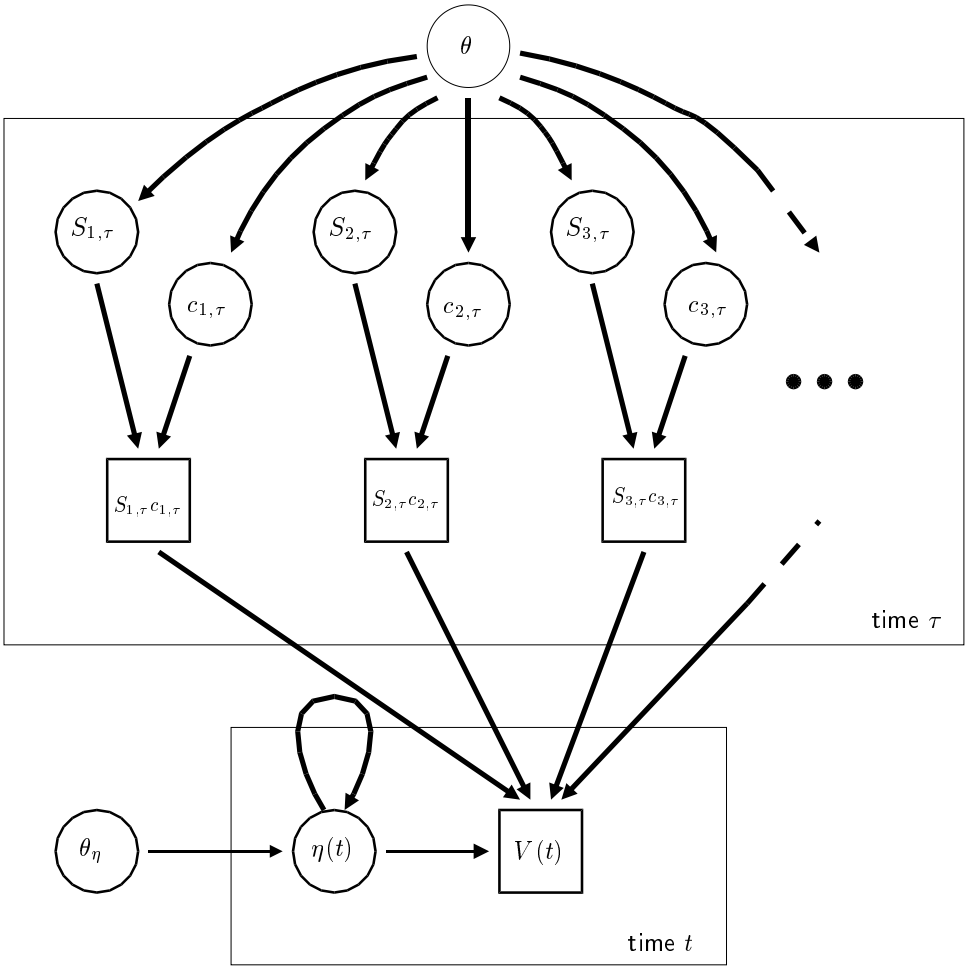

Figure 5.1: Spike sorting model schema

are indeed linear (Wehr *et al.* 1999), however we will take this fact on trust in other preparations.

The model is sketched in figure 5.1. We write $V(t)$ for the recorded potential, the only observed variable in the model. If a multichannel electrode, with tips whose listening spheres overlap (for instance, a tetrode) is used, this is a vector-valued function of time. If the multiple electrode tips are far enough apart that they cannot collect signals from the same cells (more than about 100 microns) we treat each as an independent process and model the recorded voltage traces one at a time. Our model can be written as

$$V(t) = \sum_{\tau} \left(c_{1,\tau} S_{1,\tau}(t-\tau) + c_{2,\tau} S_{2,\tau}(t-\tau) + \cdots\right) + \eta(t) \tag{5.1}$$

Here, $c_{m,\tau}$ is an indicator variable that takes the value 1 if the mth foreground cell fires at time τ and 0 otherwise. If cell m fires at τ it adds a deflection of shape $S_{m,\tau}(t-\tau)$ to the recorded potential. The functions $S_{m,\tau}$ have limited support, all of which is around 0. The effect of all the background

neural sources, along with any electrical noise which might be present, is gathered into a single term $\eta(t)$. For the multichannel electrode, both $\eta(\cdot)$ and $S_{m,\tau}(\cdot)$ are vector valued functions.

Note the subscript τ applied to the spike shape S_m. This allows for variability in the shape of the recorded action potential from a single foreground cell, over and above that due to the addition of the background noise. Such variability may arise due to changes in available membrane channels, or due to changes in the membrane surface that participates in each spike. The nature of this intrinsic variability will be discussed at greater length below. In any case, it is of a quite different character to that due to the background: it is potentially different for each cell, it need not be stationary over the course of the spike, and while we will argue below in favour of a Gaussian distribution for the background, this foreground variability is unlikely to be Gaussian in nature. The separation of the distribution of spike shapes from a single cell into these two parts is a critical feature of our approach, and one that was lacking in previous algorithms.

The random variables in our schema, as we have written it, are the background $\eta(t)$, the firing indicators $c_{m,\tau}$ and the spike shapes $S_{m,\tau}$. None of these are directly observed; however, we think of the foreground variables, $c_{m,\tau}$ and $S_{m,\tau}$ as the only latent variables in our model. We can treat $V(t)$ as a random variable, whose distribution conditioned on the latent variables subsumes the noise $\eta(t)$. The parameters of the model can be separated into two groups θ_η which governs the conditional $\mathsf{P}\left(V(t) \mid \{c_{m,\tau}, S_{m,\tau}\}\right)$ and, simply, θ governing the distribution of $S_{m,\tau}$ and $c_{m,\tau}$. Thus, we have factored the underlying distribution so:

$$\mathsf{P}\left(V(t)\right) = \mathsf{P}_{\theta_\eta}\left(V(t) - \textstyle\sum_{\tau,m} c_{m,\tau} S_{m,\tau}(t-\tau)\right) \mathsf{P}_\theta\left(\{c_{m,\tau}, S_{m,\tau}\}\right) \tag{5.2}$$

We have said nothing yet about the nature of the distributions in this factorization. This is why it is a schema and not a full blown model. We will argue that the background process is approximately zero-mean Gaussian, and the distribution of $V(t)$ conditioned on the latent variables will be normal in all of our instances of the schema. The distributions of the $c_{m,\tau}$ and $S_{m,\tau}$ will vary, and indeed, in applications will not always be the same for all foreground cells. Figure 5.1 is drawn as though all of the $c_{m,\tau}$ and $S_{m,\tau}$ were independent. This is merely for clarity in the diagram, we will consider below models for which this is not true.

Our eventual goal within each model is to infer the posterior distribution $\mathsf{P}\left(c_{m,\tau} \mid V(t)\right)$. In practice we will not carry out the marginalization over the parameters implied in that posterior; instead, we will approximate the marginal posterior by the posterior conditioned on estimated values of the parameters $\mathsf{P}\left(c_{m,\tau} \mid V(t), \hat{\theta}, \hat{\theta}_\eta\right)$. The rationale behind this approximation is explained in section section 1.2. In the next few sections we will address the problem of finding these estimates (that is, learning) within the various models that appear in our schema, as well as that of selecting an appropriate model from the schema. After this, we will turn to the question of efficient inference

of the foreground spike occurrence times.

5.4 Learning within the Schema

Separating foreground and background

The foreground and background cells in our model are distinguished entirely on the basis of the amplitudes of their spikes on the recording electrodes. It is therefore reasonable to identify the times of firing of the foreground cells using a simple amplitude threshold. We take the times at which the signal crosses the threshold (the details of which are discussed below) and extract a short segment of the signal, corresponding to the typical length of a spike waveform, around each one. These segments, which we shall refer to on occasion as **events**, contain the foreground spikes. The remaining stretches of signal are presumed to be generated by the background noise process.

This separation of foreground and background allows us to divide our learning procedure into two stages. We examine the stretches of background activity directly to estimate the parameters of the noise. Armed with this estimate, we learn the remaining parameters from the foreground events. This second stage is considerably more straightforward given an independent estimate of the background distribution. Earlier approaches, which did not differentiate between background noise and spike shape variability, did not enjoy this advantage. The choice of distribution and resulting parameter estimation for the noise will be explored in detail below.

Independent components analysis

We consider the problem of estimating the parameters θ which govern the distributions of the latent variables $c_{m,\tau}$ and $S_{m,\tau}$. On the surface, the model (5.2) is quite similar to the generative model which underlies statistical signal separation algorithms such as independent components analysis (ICA) (Jutten and Herault 1991; Comon 1994; Bell and Sejnowski 1995; MacKay 1999) or independent factor analysis (IFA) (Attias 1999). In these algorithms, signals from a group of independent non-Gaussian sources (in the spike sorting case these would be the different cells) are mixed linearly onto multiple channels of output. The output channels may then have noise, usually Gaussian, added. Learning algorithms in such models have been well studied.

Unfortunately, there are significant differences between our model and these ones. We shall note three here: two of these might be surmountable, but the third makes it very difficult to envisage such a solution in the current context.

1. ICA models generally involve exactly as many sources as output channels. If the number of cells is smaller than the number of channels this poses no problem; the algorithm would simply resolve some part of the noise as another "source", which could subsequently be discounted

using some heuristic. However, the number of cells may well be greater than the number of electrode tips that can be practically introduced. In hippocampal recordings, for example, more than 10 cells are often recorded on a single tetrode.

2. Most ICA models imply that the sources are mixed in an instantaneous manner (that is, the output at a point in time depends only on the source signals at that time). In the case of extracellular electrophysiological data, where the electrode tip properties result in filtering of the recorded signal, the mixing cannot be instantaneous. Recently, Attias and Schreiner (1998) have proposed a signal separation algorithm that resolves this difficulty.

3. The most severe difficulty is posed by the extended nature of the sources and recording surfaces. While it would seem sensible to regard each cell as a single source, the different electrode tips will, in fact, lie closest to different parts of the cell membrane, and thus record slightly different spike waveforms. As a result, one cannot treat an isolated foreground spike as a single waveform scaled linearly (or even filtered linearly) onto the multiple recorded channels. The spike waveform must itself be regarded as a fundamentally multichannel entity. This prevents the application of blind source separation techniques to spike sorting in many preparations, notably in neocortical recordings.

If we cannot use these well-established signal processing techniques, can we hope to solve the problem? In fact, ICA-like techniques fail to exploit the significant amount of prior knowledge available about the neural signal. Nowhere in the generative model for ICA, for example, is it acknowledged that a single source signal will always be a chain of approximately stereotypical pulses. It is this repetitive nature of the signal that we will exploit to solve the problem.

Before leaving this point, we make two additional observations. First, consider the following scheme for application of ICA. We regard each source as producing a train of delta-functions, with the spike waveform on each channel, however it is produced, appearing as the impulse response of a fictitious linear filter. The delta-function trains are convolved with their corresponding filters and summed (along with noise) to produce the recorded signal. The filtering and summing represent the mixing stage of a **dynamic components analysis** (DCA) model (Attias and Schreiner 1998). This treatment would seem to restore our faith in the applicability of an ICA-like algorithm. Even better, it would indeed incorporate our prior belief in the pulsatile nature of each source. The difficulty with this approach lies in the presence of spike waveform variability in the data. Since, in this scheme, the waveform information is treated as part of the mixing process rather than as a source signal, we would require a variable mixing process. Such variability cannot be handled within the DCA framework.

Second, it should be borne in mind that there may well be preparations in which ICA-like algorithms are applicable to spike sorting. For example, the form of ICA suggested in the preceding

paragraph might be successful in cases where there is little or no spike shape variability. Another example is provided by Brown *et al.* (1998) who have reported success in optical recordings of voltage-sensitive-die-treated *Tritonia* tissue. In this example, the recordings are sufficiently slowly sampled that the spread of signal across the membrane is effectively instantaneous (Brown, personal communication). As a result, the spike waveforms recorded on different photodetectors may indeed be linearly scaled versions of a single waveform. Furthermore, the optical nature of the recording ensures that the signal mixing at the detector is linear and instantaneous.

Clustering algorithms

Our approach to learning the waveform parameters is based on two observations. First, all the spikes recorded from a single cell are expected to be roughly similar. Indeed, we will specify the exact nature of the variability that we expect, by specifying the distribution of $S_{m,\tau}$ within the generative model schema. Second, the probability that two foreground cells will fire so close together in time that their spike waveforms overlap in the recorded signal is relatively low. As a result, most of the foreground events gathered by the application of our threshold represent only a single spike waveform. Thus we might expect to learn the shapes of the underlying waveforms (and the distributions of such shapes) by **clustering** these foreground events.

Consistent with our probabilistic viewpoint, we shall adopt a generative-model-based approach to clustering, as was outlined in chapter 2. To do this we need to transform the model of (5.2) into a suitable form.

Whereas (5.2) provides a model of the continuous waveform $V(t)$, we now desire a model that describes the set of extracted events, $\{V_i\}$. Each V_i is a vector of samples drawn from all of the channels of $V(t)$ around the time τ_i at which the ith event occurs. At all times τ other than the τ_i we assume that no foreground cell fired and so $c_{m,\tau} = 0$ for all m. We will employ the labels $c_{m,i}$ and $S_{m,i}$ for the latent variables at the times τ_i, in place of the more cumbersome forms such as c_{m,τ_i}.

The vectors V_i are taken to be conditionally independent, given the values of the latent variables $c_{m,i}$ and $S_{m,i}$. In other words, we assume that the separation between events is always greater than the correlation-time of the background noise process. The distribution of the ith vector is described by a **mixture** density, with one component for each possible value of the indicators $c_{m,i}, m = 1 \ldots M$. Let us consider these components one by one.

1. All $c_{m,i} = 0$. This implies that the threshold was reached by the background process alone without a foreground spike. In this case the density of the vector V_i is exactly that of the background noise, expressed as a vector density, rather than as a continuous process density.

We will introduce a new indicator variable $z_{\emptyset,i}$ to indicate this condition, and write

$$\mathsf{P}\left(V_i \mid z_{\emptyset,i}=1\right)=\mathsf{P}_{\theta_\eta}\left(V_i\right)=\mathsf{P}_{\emptyset}\left(V_i\right) \tag{5.3}$$

2. Only one of the $c_{m,i}=1$. Such events will make up the majority of those detected. We use indicators $z_{m,i}, m=1 \ldots M$ to represent each of these states (the $z_{m,i}$ are exactly the same as the corresponding $c_{m,i}$, though only in this condition). The density of the event vector is then

$$\mathsf{P}\left(V_i \mid z_{m,i}=1\right)=\int dS_{m,i}\, \mathsf{P}_{\theta_\eta}\left(V_i-S_{m,i}\right) \mathsf{P}_\theta\left(S_{m,i} \mid \{S_{n,j}, c_{n,j}: j<i\}, c_{m,i}=1\right) \tag{5.4}$$

Notice the conditioning of $S_{m,i}$ which depends only on the preceding latent variables to enforce causality. We will abbreviate this set of latent variables at all times earlier than τ_i by $\lambda_{<i}$ and write this density as $\mathsf{P}_m\left(V_i \mid \lambda_{<i}\right)$.

3. More than one $c_{m,i}=1$. In this case two foreground cells fired at close enough times that the threshold was only crossed once by the compound waveform. We expect such events to occur rarely *and will not explicitly model them as overlapped events at this stage.* Instead, we treat all such waveforms as "outliers", and model them by a single, uniform density (see section 2.7.1). We introduce a latent variable $z_{\not\emptyset,i}$ to indicate this condition. The corresponding density is simply

$$\mathsf{P}\left(V_i \mid z_{\not\emptyset,i}=1\right)=\begin{cases}\frac{1}{\|A\|} & \text{if} \quad V_i \in A \\ 0 & \text{if} \quad V_i \notin A\end{cases} \tag{5.5}$$

with A some region of the vector space of V_i and $\|A\|$ its volume. We will write this density as $\mathsf{P}_{\not\emptyset}\left(V_i\right)$.

The complete model for the ith vector is thus

$$\begin{aligned}\mathsf{P}\left(V_i\right) &= \mathsf{P}_\theta\left(z_{\emptyset,i}=1 \mid \lambda_{<i}\right) \mathsf{P}_{\emptyset}\left(V_i\right) \\ &+ \mathsf{P}_\theta\left(z_{\not\emptyset,i}=1 \mid \lambda_{<i}\right) \mathsf{P}_{\not\emptyset}\left(V_i\right) \\ &+ \sum_{m=1}^{M} \mathsf{P}_\theta\left(z_{m,i}=1 \mid \lambda_{<i}\right) \mathsf{P}_m\left(V_i \mid \lambda_{<i}\right)\end{aligned} \tag{5.6}$$

Once again, the distribution of the indicator variables is conditioned only on earlier latent variables so as to preserve causality in the model.

The latent indicator variables $z_{m,i}$, $m=\emptyset, \not\emptyset, 1 \ldots M$ are mutually exclusive: exactly one of them takes the value 1 for any i, while all of the rest are 0. As such, they closely resemble the mixture latent variables of chapter 2. In many of the models we will discuss, the indicators for each event will be drawn independently from a fixed distribution. In this case, the model is exactly a mixture

model. Even where this is not exactly true, however, we shall call this the **mixture form** of the generative model. Fitting such a model is what we will mean when we claim to be performing a parametric clustering of the spike events.

It is worthwhile to consider the impact of our choice not to model the overlapped spike events explicitly, but rather to sweep them into a single outlier distribution. Is it likely that this inaccuracy in the event model (5.6) will lead to estimates of the parameters that do not carry over to the true continuous signal model (5.2)? The mistreatment of overlaps poses two distinct dangers to accurate parameter estimation. The first is that some overlaps will be incorrectly interpreted as single spikes, and thus bias the estimate of the spike shape distribution of the misidentified cell. This possibility is slim. Overlaps need to be fortuitously exact to look anything like single spike waveforms. Most likely, they will fall quite far from any single cell cluster and be easily recognized as outliers. Furthermore, the use of a uniform outlier distribution minimizes the expected bias in estimates of the mean spike shapes of each cell (robustness to outliers in the fitting of mixture models is discussed in section 2.7.1). The second danger arises from the fact that the occurrence of an overlap "removes" an event which would otherwise contribute to the parameter estimation. For models in which the latent variables associated with each event are independent of all others (these are the true mixture models) this effect will be negligible, provided that the probability of overlap is small and independent of the latent variable values. However, for models in which the spike shape and probability of firing for each cell depend on its history, this can pose a real problem. We shall address it when we discuss such models.

For the sake of the reader familiar with previous spike sorting techniques it is worth emphasizing here a point that has appeared before, and will be addressed again in section 5.14. In the present approach to the problem, unlike in many (though not all) others, the clustering stage is a preliminary to the inference of spike arrival times. We use it as a device to learn the parameters θ that govern the distributions of $c_{m,\tau}$ and $S_{m,\tau}$. The actual inference of the variables $c_{m,\tau}$ is done within the more accurate superposition model (5.2), without the imposition of an artificial threshold, nor the rejection of overlapped spikes.

5.5 Event Detection

Our first step in the process of learning the model parameters is to identify the times at which foreground cells fired by comparing the recorded signal to a threshold amplitude.

A short segment of data recorded from the neocortex of a macaque monkey using a tetrode is shown in figure 5.2A (the four traces show the simultaneously recorded signals on the four wires). Large amplitude spikes are clearly superimposed on a lower amplitude background process. However, it is clear that the comparison of this raw signal to a fixed threshold will not achieve the separation

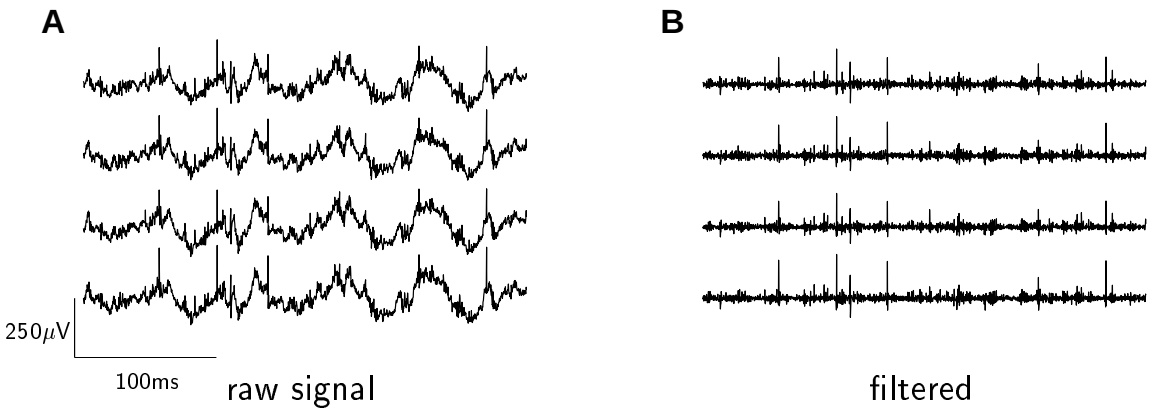

Figure 5.2: A sample extracellular recording.

we desire; the signal exhibits a low frequency baseline modulation with an amplitude comparable to that of the largest foreground spikes. This low-frequency field potential signal may be of considerable interest in itself, however the frequencies involved are too low to have an influence on the shapes of the relatively short spike waveforms and so it can safely be removed for the purposes of spike sorting. Figure 5.2B shows the same segment of data after it has been digitally high-pass filtered. The filter cutoff is chosen at the lowest frequency that can contribute to the foreground spike shapes, based on the length of those spikes. For neocortical recordings of the type shown in figure 5.2 the spike length is not longer than 2 milliseconds, implying a filter cutoff of at least 500Hz.

We wish to choose a threshold which allows us to identify the spikes that rise above the background process. To do this we need to know the statistics of the background, but, of course, we cannot measure these until we have separated background from foreground. We shall set the threshold in terms of the variance of the entire signal, foreground and background. In doing so, we assume that foreground spikes are rare enough that this measurement is dominated by the background. This may not always be true: if we record 4 foreground cells, all firing at about 50Hz, there would be a total of 200 spikes in one second of recording. As the large amplitude peak of each foreground spike can last up to half a millisecond, this would mean that one-tenth of the recording has large amplitude foreground contributions – enough to affect the background variance estimate. As a result, a certain degree of user intervention is useful in setting the threshold level. A typical choice of threshold is 3–5 times the root-mean-square value of the high-pass filtered signal.

Spike waveforms are generally biphasic pulses. The strongest currents during an action potential are associated with the influx of sodium that initiates the firing; as a result, the first phase is almost always the larger. The sodium current flows into the cell, away from the electrode tip. Thus, this first phase is negative on the electrode. Under the polarity convention adopted in this chapter (introduced in section 5.2) it will appear positive in our recordings. In order to reduce the probability

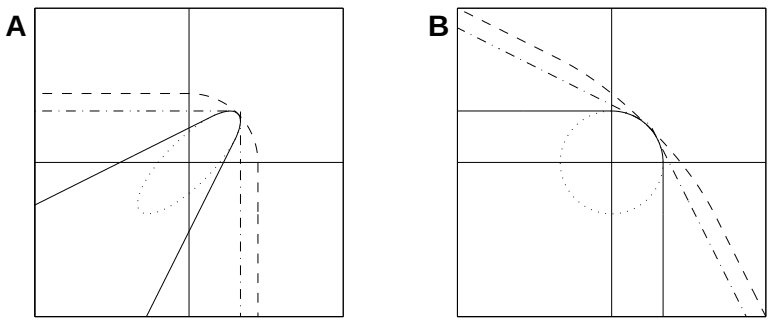

Figure 5.3: Event detection thresholds

of false triggers, and also to ensure that each spike causes only one threshold crossing, we apply the threshold in a one-sided manner, accepting only crossings where the recorded potential rises above the positive threshold value.

It is not obvious how to apply the threshold to multichannel data. We shall consider three schemes here, and it will be useful to compare them graphically. This is done for a hypothetical two-channel signal in figure 5.3. The axes in panel A represent the amplitude of the signal on the two channels: each sample of the signal is represented by a point in this plane. The thresholding schemes will be described by boundaries in the plane which separate regions where the signal is below the threshold from regions where it is above. The various lines in this panel, and the nature of panel B, will be described below.

The most commonly employed approach to multichannel data is to accept an event whenever any one channel rises above a scalar threshold. The acceptance boundary of such a threshold for the two-channel example is represented by the dash-dotted line in figure 5.3A. The signal has crossed this simple threshold if the point falls to the right of or above the line. We shall call this a **rectangular** threshold.

An alternative approach would be to threshold the total instantaneous power of the signal, that is, the sum of squares of the amplitudes on the various channels. Given the unidirectional nature of the spike peaks, we choose to half-wave rectify the signal before squaring. The resultant threshold, which we call **circular**, is shown by the dashed line.

The dotted ellipse in figure 5.3A shows a covariance contour for the background distribution, that is, a line drawn at a constant distance from 0 in the Mahanalobis metric defined by the distribution's covariance. The ellipse is drawn as though the background on the two channels is positively correlated. In fact, this is the overwhelmingly dominant case in experimental data. It is reasonable that electrode tips close enough to share spikes from the same foreground cells will also share background spikes.

A comparison between this elliptical noise contour and both of the threshold boundaries described

so far reveals the weakness in these approaches. Many points above and to the right of the ellipse are unlikely to arise purely from the background process, and yet are not detected as foreground events. A more sensible approach would seem to be to shape the boundary to match the contour of the second moment of the noise distribution. This is conceptually easiest in the noise-sphered space, which is obtained by an instantaneous linear transformation on the signal (if the noise covariance is Σ the sphering matrix is $\Sigma^{-1/2}$). This space is represented in figure 5.3B. The noise covariance matrix is now, by construction, spherical. The rectangular and circular thresholds are shown in the dot-dashed and dashed lines, as before. The solid line represents a threshold boundary constructed in the same way as the circular threshold, but now in the sphered space; the solid line in panel A shows the shape of this boundary in the original space. We refer to this as the **elliptical** threshold.

By construction, the elliptical threshold matches the covariance contour of the noise. If that noise is Gaussian distributed, this curve is also an iso-probability contour, so that the probability of the noise alone exceeding the threshold is independent of the direction (in the space of figure 5.3A) of the signal.

5.6 The Background Process

Once the times of the foreground events have been identified, we explore the statistics of the signal during the periods between these events, with the goal of characterizing the background process. In the first instance, we are interested in the distribution $\mathsf{P}_{\theta_\eta}(V_i)$ which expresses the background as a vector-output process. This distribution will be of critical importance in what follows: not only is it the distribution of the noise (5.3), it also makes a significant and common contribution to the distribution of spike waveforms recorded from each cell (5.4).

We estimate the distribution of the V_i directly, by sampling the background process at times when no foreground spike is present. The spikes extend for some time before and after the times of the threshold crossings; thus, we need to extract vectors away from these points so as not to overlap the foreground waveforms. For the data shown here, no samples were taken within 1.6ms of each crossing. The remaining signal is then broken up into segments whose length matches the duration of a foreground spike. Each such segment represents a single vector sample of the background process. We study the distribution of the ensemble of these vectors along the principal components.

Each of the columns of panels in figure 5.4 shows the density of the loadings of the noise vectors on a selection of the ensemble principal components, for an example macaque tetrode recording. In each column the upper and lower panels show the same data; the upper panel shows the density directly, while the lower panel shows the log density, thereby revealing the details of the tails of the distributions. The rank of the component on which the loadings are taken is indicated below the column. The dots represent the density histogram of the observed vectors. The continuous line

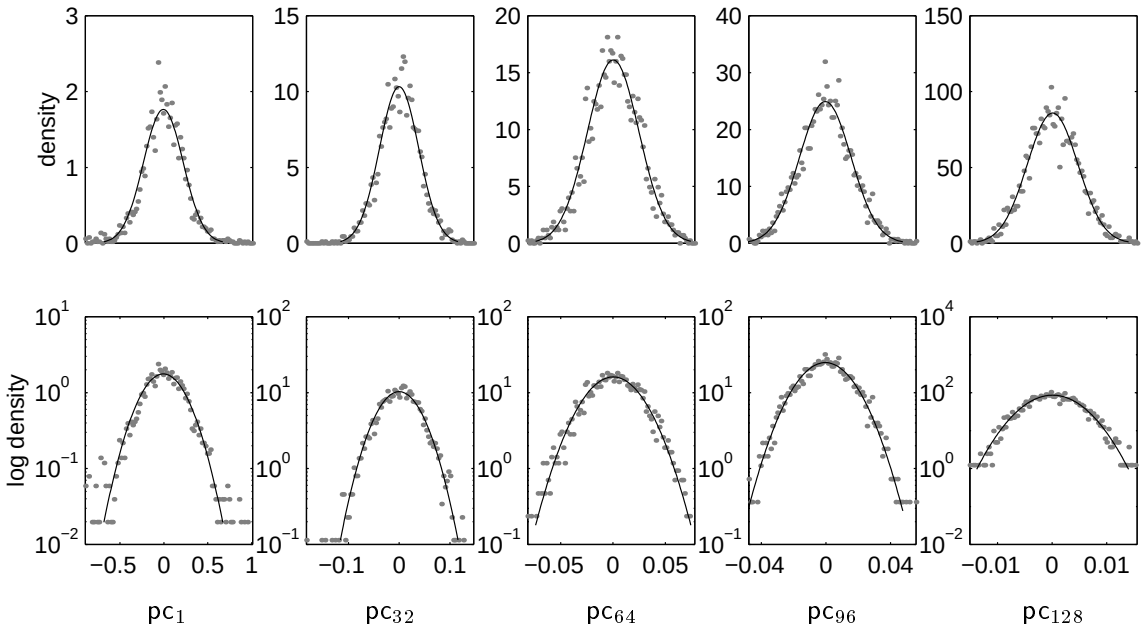

Figure 5.4: The distribution of background noise

represents a Gaussian density with the same variance as that of the observed loadings. It is clear that a Gaussian model for the background process is reasonable, although a slight excess in kurtosis is evident in the first components.

In the rest of this chapter we shall take the background to be Gaussian distributed. While figure 5.4 suggests that this is reasonably well supported by the data, it is not exactly true in all cases (Fee *et al*. 1996b). Our choice is driven by two observations. First, the Gaussian model considerably reduces the computational demands of the various approaches that we will discuss, and is quite important for efficient separation of overlapped spike waveforms. Second, we will introduce separate models for intrinsic spike variability that will be non-Gaussian. Thus, it is possible for some non-Gaussian background noise to be subsumed by these models. In situations where computational cost is no object, or where the data exhibit extreme departures from normality, an alternative distribution may be used for the background. Most of the generative models to be discussed will carry through with little modification. The largest cost will come in the final stages of spike-time inference, where the filtering scheme we adopt is critically dependent on Gaussian noise.

A zero-mean Gaussian density is entirely specified by its covariance matrix. Since the background process is stationary with respect to the duration of the spike waveform — that is, the statistics of the background are the same at each point along the spike — this covariance matrix may be constrained to have Töplitz (diagonally striped) structure. Thus, the only parameters of the noise distribution are given by the autocorrelation function of the background.

While the noise is almost certain be stationary on the time-scale of a single spike waveform,

it may well be appreciably non-stationary on time-scales of hundreds of milliseconds or more. In particular, as stimulus conditions change, the rate of firing of both foreground and background cells will change, quite probably in a correlated fashion. Thus, by sampling the background far from the locations of the foreground spikes we run the risk of measuring a background quite different from that which actually affects the distribution of event waveforms.

We can avoid this pitfall by biasing the sample of background vectors so that most are drawn close to, though not overlapping with, the foreground spikes. One simple procedure to ensure this it to sample a fixed offset from each foreground spike (after making sure that this would not result in an overlap with a different event). Another is to sample exactly in-between each pair of adjacent events (again making sure that the pair is far enough apart that this will not cause an overlap). Furthermore, in extended recording we can re-estimate the noise continuously, leading to an adaptive estimate that can track non-stationarities on the time-scale of seconds.

5.7 Foreground Events

Models within the mixture schema (5.6) describe a multivariate density for foreground events. In this section we shall examine the procedure by which a vector representation in constructed for each foreground spike. We proceed in two steps: in the first the vector elements are sampled directly from the voltage trace yielding relatively high-dimensional vectors; in the second we use a low-rank linear transform to reduce this dimensionality through a technique similar to principal components analysis.

5.7.1 Extraction and alignment

In the first stage, each element of the event vector will be a sample drawn from the recorded voltage trace near the time of the corresponding threshold crossing. The extracted samples will be separated by the Nyquist sampling period derived from the frequency content of the signal, which in turn is controlled by an analogue anti-aliasing filter. We order the samples forward in time, with all of the samples from the first channel appearing together, followed by the samples from the second channel if there is one, and so forth. In multichannel recordings, the corresponding samples on each channel will always be simultaneous.

A common approach to selecting the vector coordinates is to copy a fixed number of values from the digitized recording before and after the sample at which the threshold was crossed. This, however, does not ensure that the samples are taken at the same time relative to the underlying spike waveform. This jitter in sampling introduces artificial variability in the extracted set of vectors as illustrated in figure 5.5. Panel A shows one channel of a small number of recorded spike waveforms, all originating from a single cell. The samples extracted from the waveforms are shown by the dots;

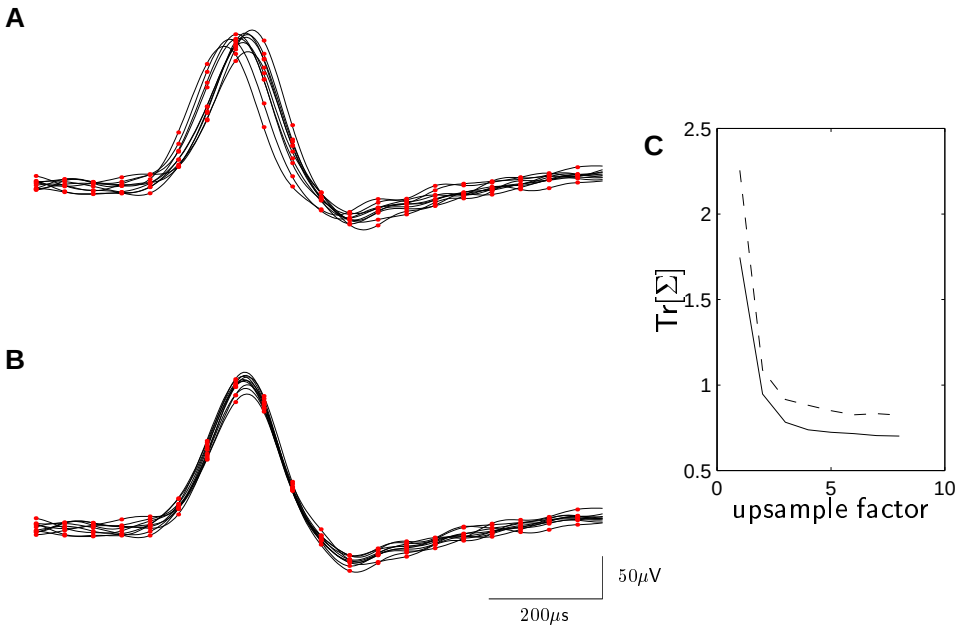

Figure 5.5: Alignment of spike waveforms.

the solid lines beneath show the Fourier reconstruction of the underlying signal, assuming the there was no power above half the sampling frequency. The variation in alignment of the underlying waveform is evident, and results in "noise" in the samples that can reach up to half of the spike amplitude. Furthermore, if the temporal jitter of the alignment is uniformly distributed within one sample interval, this apparent "noise" will also be almost uniform (its exact shape is set by the derivative of the underlying spike shape), making it difficult to model. Fortunately it can be mostly eliminated.

There are two sources of jitter. For the sake of argument, let us assume that the underlying spike waveform being measured has no intrinsic variability. In that case, there is a well defined time at which the waveform crosses the threshold, and we would like to align the samples in the event vector with this time. The first source of jitter is the background noise, the addition of which to the recorded spike waveform will result in that waveform crossing the threshold at a slightly different point from our reference time. The second source comes from the sampling of the waveform, which is unlikely to be aligned with the spike and thus the crossing-time will probably fall between two samples, rather than on one.

The jitter and its associated artifact can be reduced considerably by some amount of signal processing. The effect of the background on alignment can be reduced by choosing to align to a composite landmark, rather than a single sample level. We will use the "centre of mass" of the peak of the waveform, that is, the quantity $\tau_c = \int dt\ tS(t) / \int dt\ S(t)$ with the integrals limited to

the peak region of the spike waveform $S(t)$. This is estimated from sampled data S_n by a form similar to $\hat{\tau}_c = \sum t_n S_n / \sum S_n$, with the range of the sum limited to samples near the peak of the waveform. The sum over samples reduces the effect of the background on the alignment time. Temporal correlations in the background will interfere with this reduction, and so it is preferable to use the background-whitened signal (see section 5.6).

We can eliminate the sample-alignment jitter by resampling the waveform to align with the estimated centre of mass exactly, even if that estimate falls off the original sample grid. This resampling is achieved by interpolation, either with cubic splines, or "exactly" using Fourier techniques. The cubic spline interpolation is straightforward and will not be described here. The Fourier technique proceeds as follows. Conceptually, we find the discrete Fourier transform of the sampled waveform and treat the coefficients thus obtained as the coefficients of a finite Fourier series. Provided that the original signal was sampled at or above the Nyquist sampling frequency for its bandwidth, this series sums to the original, continuous signal (barring boundary effects). We draw new samples from this exact interpolant. The Fourier process described is equivalent to a kernel smoothing of the discrete sequence treated as a sum of delta-functions, where a sinc-function is used for the kernel. As might be expected from a sinc-function kernel, the interpolant will tend to ring near the boundaries of the interpolated segment; it is important, therefore, to use a segment sufficiently long that the region of interest does not fall critically close to a boundary.

The selection procedure for the samples to be used in calculation of the centre of mass has not yet been discussed. It proceeds as follows. First, the maximum sample within a short time after the detected threshold crossing is identified. In the region of this sample the waveform is "upsampled" by resampling from the interpolant at a higher rate. The region used extends sufficiently far on each side of the maximum to encompass the entire first peak of the spike waveform. Next the contiguous region of samples that encompassed the maximum and lies above a threshold value is identified. This threshold is chosen lower than the trigger threshold, so as to ensure that a large number of samples will fall above it. The threshold-based centre of mass calculation is preferred to use of a fixed number of samples around the maximum because it avoids the bias towards the centre of selected interval that is inherent in the latter approach.

The centre of mass is calculated by,

$$\hat{\tau}_c = \frac{\sum t_n (S_n - a)}{\sum (S_n - a)} \tag{5.7}$$

where the sums range over the contiguous samples S_n of the upsampled waveform that lie above the threshold a. The subtraction of the threshold from the sample values ensures that samples near the boundary of the selected region have little effect on the estimate, thereby protecting it from noise-driven variations in that boundary. A fixed number of samples, sufficient to encompass the

extent of the spike waveform, spaced by the Nyquist period and aligned with $\hat{\tau}_c$, are extracted from each channel of the recording and arranged into the event vector.

The results of this alignment procedure are shown in figure 5.5B. Clearly, the apparent noise has been reduced considerably. Given a group of waveforms known to originate from the same cell, we can measure the effect of the alignment procedure by calculating the trace of the covariance matrix of the spike waveforms after alignment. These values of are shown in figure 5.5C for a number of different algorithms. The dashed line represents alignment to the threshold crossing, while the solid line represents alignment to centre of mass. Furthermore, each reference point was extracted using varying degrees of upsampling (that is, interpolation). Two observations are clear: both techniques improve at about the same rate as finer upsampling is employed; and furthermore, the centre of mass reference point provides a constant benefit over the threshold crossing at all upsampling factors. The two different sources of jitter, along with the effectiveness of the proposed techniques in overcoming them, are evident.

5.7.2 Dimensionality reduction

The number of samples that goes into each vector might be quite large. For tetrode recordings in monkey neocortex, for example, a 10kHz signal bandwidth is suitable, spikes last over a millisecond in time, and so the vectors will contain more than 80 elements. Such large vectors lead to two difficulties. One is purely computational: calculations on lower-dimensional objects would be much faster. This is a particularly relevant concern for the case of on-line spike sorting. The second is perhaps more serious. As the dimensionality of the modeled space grows so does the number of parameters, and so the quantity of data needed to obtain good estimates can become very large. With insufficient data, the danger of over-fitting is considerable.

Fortunately, it is possible to reduce the dimensionality of the space efficiently and without any loss of useful information. In this discussion we will only consider linear dimensionality-reducing transforms. That is, we will seek a rectangular matrix, R, by which we can multiply the data vectors, V_i so as to obtain the lower-dimensional products $x_i = RV_i$. The x_i must retain as far as possible those features of the data set V_i which are essential to clustering.

Hand-picked features

Perhaps the most commonly adopted approach is to derive from each waveform a small group of features which might *a priori* be expected to carry much of the relevant information. For a multi-channel electrode, the most natural such features are the peak potentials attained on each recording surface. For tetrodes, then, each x_i becomes a point in $\mathbb{R}^4$. Figure 5.6 shows the events extracted from one tetrode recording, projected into this basis. The 4-dimensional space is represented by the 6 possible 2-dimensional axial projections. Thus, in the topmost panel the peak value on channel 2

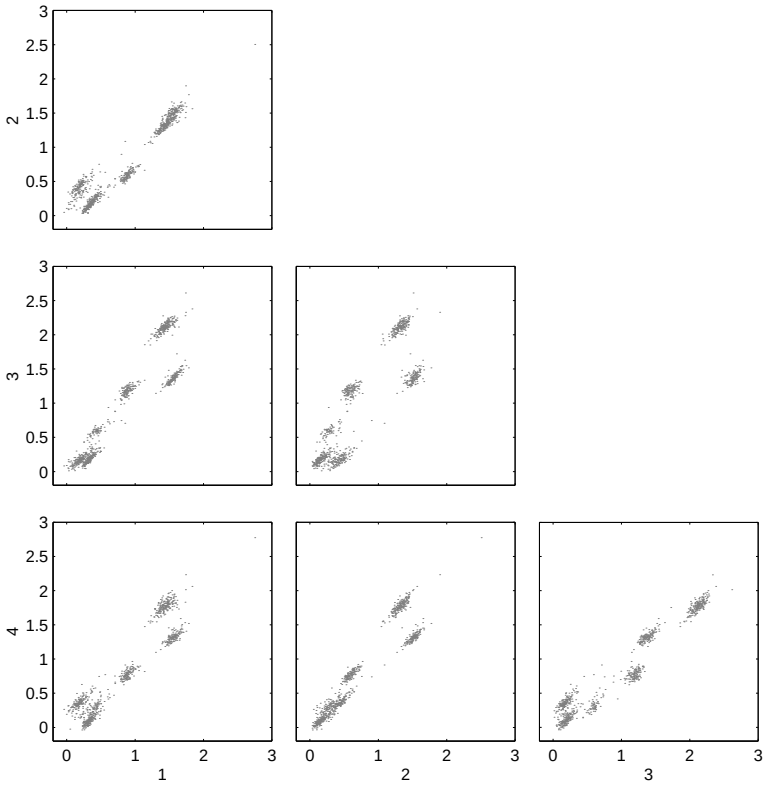

Figure 5.6: Events represented by peak voltage on four channels.

is plotted against the peak value on channel 1; in the panel immediately below, the peak on channel 3 is plotted against the peak on channel 1; to the right of this panel, the peak on channel 3 is plotted against that on channel 2 and so forth. A similar representation will be used many times in the following pages. While in figure 5.6 the numbers that appear below and to the left of the panels represent channels numbers, in many of the later diagrams they will indicate arbitrary basis vectors in the space of the events V_i.

Six distinct clusters are visible to the observed in the data of figure 5.6. However, the three closest to the origin, containing relatively low-amplitude spikes, are somewhat difficult to distinguish. Nevertheless, in this case, fitting a mixture model in this restricted subspace is likely to be quite effective.

In many cases we can reasonably define the "peak" on a given channel to be the value of a particular sample in the suitably aligned event waveform. In this case, the feature subspace can be obtained by a linear projection with a matrix R that contains mostly 0s, with a single 1 per row selecting the appropriate sample. This was the definition used to generate figure 5.6. Some other features in general use (such as the peak-to-trough amplitude) may also be extracted by linear projections. However, others, such as the width of the waveform peak, can not.

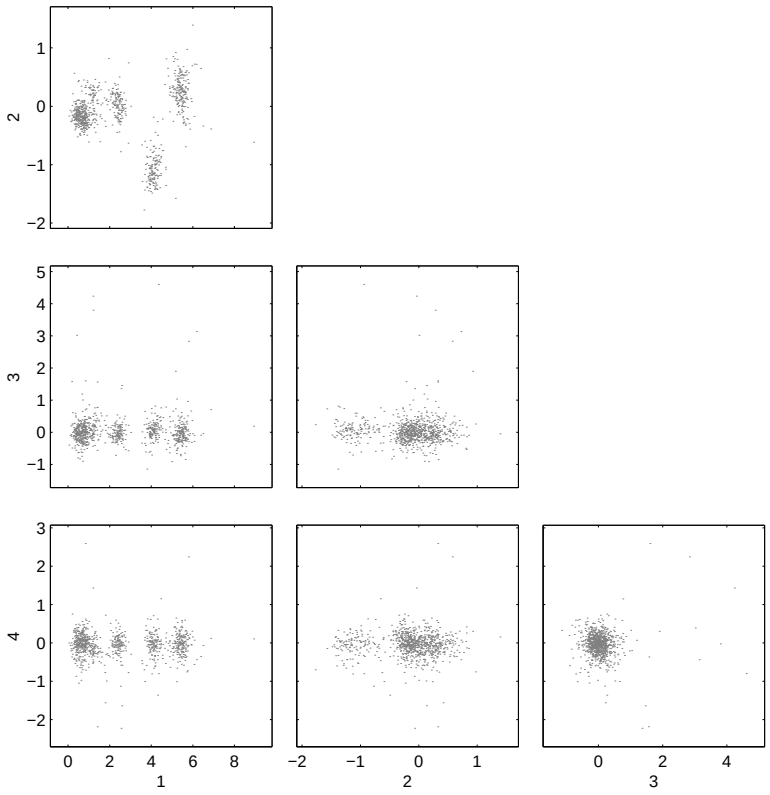

Figure 5.7: Events represented in the principal component subspace.

The attraction of linear dimensionality reduction is not simply a matter of algorithmic simplicity. A key feature of the model schema of section 5.3 is the single, consistent model for the contribution of the background process to the variability in the recorded waveforms. This simple fact remains true under any linear transformation of the space, indeed, the background model remains Gaussian to the extent described in section 5.6. Under a non-linear transformation such as spike-width, not only do we lose the Gaussian representation for the background contribution, but the contribution to the variability of this feature will be different for different underlying waveforms. This would violate the mixture schema of (5.6), making the task of statistical modeling far more difficult.

Principal components analysis

A linear approach, commonly used in situations such as this, is known as principal components analysis (PCA). PCA selects a subspace spanned by a small number of eigenvectors of the observed (total) covariance matrix

$$\Sigma_T = \frac{1}{N} \sum (V_i - \bar{V})(V_i - \bar{V})^T \tag{5.8}$$

The eigenvectors chosen are those with the largest associated eigenvalues. The resultant projection has the property that, among all the linear projections of the same rank, it retains the greatest

amount of the original data variance. We expect the PCA projection to be useful because clustering is likely to be easiest in those directions in which the data are well spread out. However, it may not be the optimal projection.

Figure 5.7 shows the projection into the first four principal components (in order) of the same data set as was shown in figure 5.6. In this case, our expectation that PCA will improve the separation of the clusters is belied. Where six different groups could be made out in figure 5.6, only four can be clearly resolved here. Furthermore, the clusters are separated in only the first two dimensions. This experience is not uncommon when handling tetrode data.

The optimal linear projection

It is well known that we can obtain the optimal linear projection *a posteriori*, that is, given knowledge about which cell each spike originated from. The procedure, known as linear discriminant analysis (LDA), selects the linear projection in which the separability of the clusters is maximized, that is, the ratio of the average distance between the clusters to the average spread of the data within each cluster is greatest.

We introduce two new covariance or scatter matrices, the between-class scatter Σ_B and the within-class scatter Σ_W. Let us identify the vectors that fall in the mth class by $V_{m,i}$, and write the mean of all such vectors as $\bar{V}_m$, with $\bar{V}$ being the overall mean as before. The number of vectors in the mth class will be written N_m, and the fraction of the total that this number represents, π_m (these fractions being equivalent to the mixing probabilities of a mixture model). The two new scatter matrices are defined thus

$$\Sigma_B = \sum_m \pi_m (\bar{V}_m - \bar{V})(\bar{V}_m - \bar{V})^T \tag{5.9}$$

$$\Sigma_W = \sum_m \pi_m \frac{1}{N_m} \sum_i (V_{m,i} - \bar{V}_m)(V_{m,i} - \bar{V}_m)^T \tag{5.10}$$

The symmetrized ratio we wish to see maximized in the projected space is $\Sigma_W^{-1/2} \Sigma_B \Sigma_W^{-1/2}$. Just as in PCA, we find the eigendecomposition of the corresponding matrix in the higher dimensional space and then project onto the space formed by the leading few eigenvectors.

It would appear that we can obtain little advantage from the discriminant approach, as the scatter matrices given by (5.9) and (5.10) cannot be calculated without access to the very information that we seek. However, it is possible to view the LDA procedure in a different light. Consider a transformation of the vectors $V_{i,m}$ by the matrix $\Sigma_W^{-1/2}$ to obtain new vectors $\tilde{V}_{i,m}$. Direct substitution into (5.10) reveals that in this transformed space, the within-class scatter, $\tilde{\Sigma}_W$, is the identity matrix. We shall refer to this as the **class-whitened** space. To now perform LDA, we need only maximize the between-class scatter $\tilde{\Sigma}_B$. It is straightforward to see that the subspace thus identified

is exactly the same as would be obtained by discriminant analysis in the original space. Indeed, this whiten-and-diagonalize algorithm is a common implementation for LDA (see, for example, Ripley (1996)). We can go one step further if we note that the total covariance in the class-whitened space is simply $\tilde{\Sigma}_T = \tilde{\Sigma}_B + \tilde{\Sigma}_W = \tilde{\Sigma}_B + I$. Thus the overall scatter matrix is diagonalized in the same basis as the between-class scatter matrix. LDA is equivalent to PCA in the class-whitened space.

The key point of this analysis is the simple relationship $\Sigma_T = \Sigma_B + \Sigma_W$. This implies that we need only one of the classification-dependent scatter matrices in order to find the optimal discriminant subspace, the other can be derived from the overall variance of the data. We do not know either of these matrices, but we do have an (under)estimate of the average within-class scatter Σ_W, provided by the direct measurement of the background. Thus, we can find a basis quite similar to the optimal LDA basis by taking the principal components in the **noise-whitened** vector space. An example of this procedure will appear in figure 5.8.

Robust principal component analysis

Inevitably, some events within the ensemble will fall far from any clusters. These are mostly the events that contain overlapped spikes as described in section 5.4. Since the data covariance matrix weights points by the square of their distance from the mean, principal components calculated from the entire data set are particularly sensitive to the number and location of these outliers. It is important, therefore, to obtain the components in a manner that is robust to outliers.

We will adopt an approach to robustness similar to that discussed in the context of the clustering algorithms in section 5.4. We can view the PCA procedure as fitting a multivariate Gaussian distribution to the data and then selecting a projection on the basis of the fit distribution. This relationship between PCA and Gaussian modeling has been explored quite extensively in the recent past (Tipping and Bishop 1997; Roweis 1998). Following the argument made during the discussion of the impact of outliers on clustering, we replace the single Gaussian by a mixture of a Gaussian and a uniform density (the limits of the uniform density being set by the maximum extent of the data). Recall from the discussion of section 5.4, that the introduction of the uniform component will not, on average, bias the estimates of the eigenvectors of the covariance of the Gaussian component. It is these eigenvectors which represent the principal component basis.

Figure 5.8 shows the subspace obtained when this robust PCA is applied in the noise-whitened space. The six clusters are now very much in evidence, and comparison with figure 5.6 suggests that they are better separated. Figure 5.9 shows the data set projected into the first four dimensions of the optimal linear discriminant space, calculated *a posteriori* from a mixture fit to these data. Clearly, for this recording, the noise-whitened robust PCA technique has identified a subspace remarkably close to the optimal one.

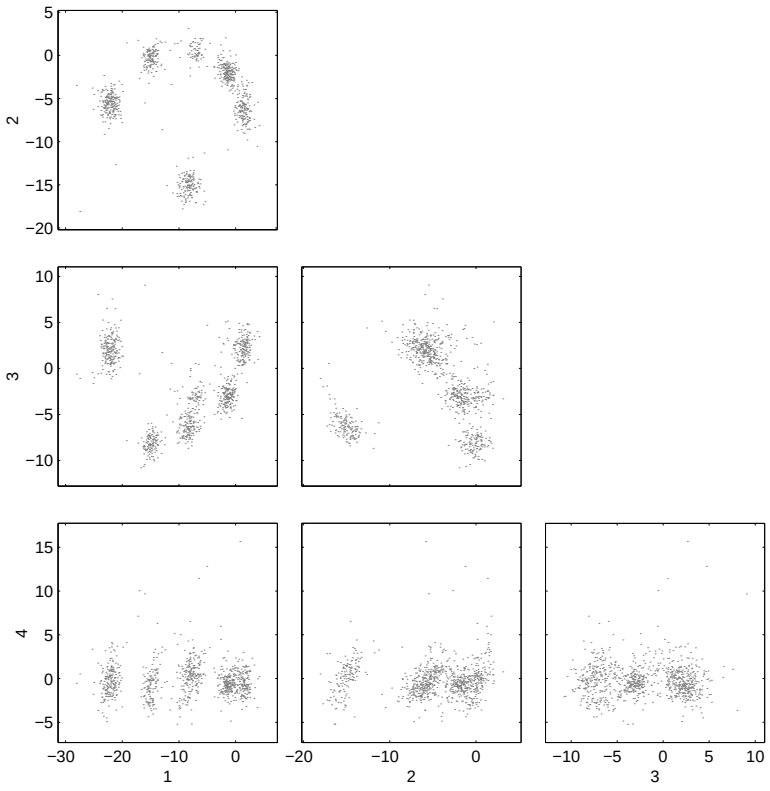

Figure 5.8: Events represented in the noise-whitened robust PCA subspace.

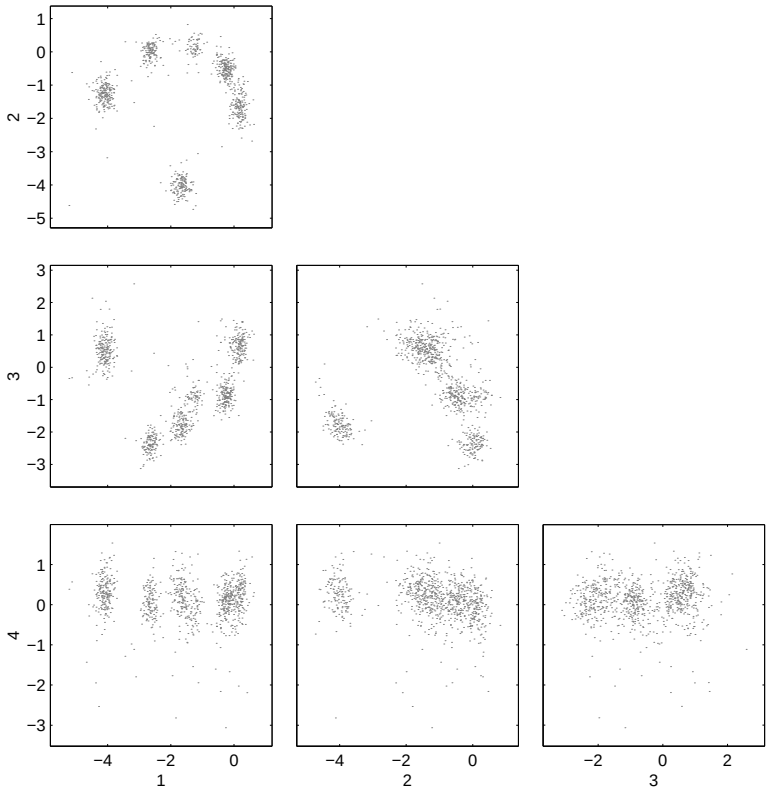

Figure 5.9: Events represented in the optimal linear discriminant space.

Outlier rejection

Dimensionality reduction carries with it the danger of reintroducing outliers into the main body of the ensemble. The danger arises in the case of outliers which fall outside the principal distribution along the directions which are to be suppressed, but whose projections onto the preserved space are not easily distinguished from those of normal spikes. Such outliers may bias the estimation of waveform parameters. Fortunately, they can be eliminated by removing from the ensemble spikes which exceed a data-set threshold in the suppressed directions. If the robust principal components analysis is used, they may be identified as points for which the uniform outlier component takes significant responsibility.

5.8 The Simple Mixture Model

5.8.1 The model

Once the ensemble of vectors has been extracted, we proceed to fit a model drawn from the schema (5.6), with the observations V_i replaced by the processed, lower dimensionality vectors, x_i. Initially, we shall examine the simplest possible such model.

We begin with two assumptions. First, each measured event vector is taken to be independent of all the others. This implies both that the set of indicators $\{z_{\emptyset,i}, z_{\not\sigma,i}, z_{m,i}\}$ are independent for different i (clearly, for any given i, they cannot be independent as only one can take the value 1) and also that the spike shape measured depends only on which cell fired, not on the previous waveforms emitted by that, or any other, cell. This assumption, allows us to drop the conditioning on the past latent variables (which was written "$\mid \lambda_{<i}$" in (5.6)). We write π_r for $\mathsf{P}_\theta\left(z_{r,i}=1\right)$ for $r = \emptyset, \not\sigma, 1 \ldots M$.

Second, the intrinsic variability in the spike shape is taken to be negligible, so that all of the observed variation is due to the addition of random background noise. In this case, each of the spike waveform densities $\mathsf{P}_m\left(x_i\right)$ is a Gaussian, whose mean is the spike shape associated with the mth cell and whose covariance is that of the background process. For noise-whitened data, this is the identity matrix.

Combining these assumptions with the mixture model schema (5.6), and restricting to the reduced-dimensionality space of the x_i, we obtain the basic model

$$\mathsf{P}\left(x_i\right) = \pi_\emptyset \left|2\pi I\right|^{-1/2} e^{-\frac{1}{2}\|x_i\|^2} + \sum_{m=1}^{M} \pi_m \left|2\pi I\right|^{-1/2} e^{-\frac{1}{2}\|x_i-\mu_m\|^2} + \pi_{\not\sigma}\mathsf{P}_{\not\sigma}\left(x_i\right) \tag{5.11}$$

where $\mathsf{P}_{\not\sigma}\left(x_i\right)$ is the uniform density given in (5.5).

5.8.2 Parameter estimation

Such a model is easy to fit. We employ the well-known Expectation–Maximization (EM) algorithm (Dempster *et al.* 1977; see chapters 1 and 2 of this dissertation) to find the maximum-likelihood parameter values. Other techniques, such as gradient-ascent or Fisher scoring may also be used for optimization. EM, however, offers some advantages.

1. EM is, perhaps, the most flexible of the various hill-climbing techniques, being easily extended to the more complex models to be discussed below. As a result, it provides a uniform approach to the fitting of the various models within the schema. Further, it is easily adapted to the situation in which different generative distributions are used for different cells, which will be discussed in section 5.11.

2. Incremental variants of EM are provably correct (Neal and Hinton 1998). While such proofs are derived in the case of static parameter values, they can give us confidence that similar variants will be well-behaved in the case of slowly drifting parameters, allowing us to track such drift.

3. The EM algorithm is very closely linked to the maximum-entropy deterministic annealing clustering technique (Rose *et al.* 1990). Indeed, the deterministic annealing approach can be extended to any latent variable model where EM is used by the Relaxation EM (REM) algorithm of chapter 3 (see also Ueda and Nakano (1998)). This technique provides a initial-condition-independent optimum, relatively immune to local maxima.

The EM iterations for simple mixture models such as this were derived in section 2.4. The current model has some additional constraints which further simplify the fitting procedure.

The background component distribution in (5.11) is fixed; only the mixing parameter $\pi_\emptyset$ needs to be learnt. The uniform outlier distribution has parameters that describe the region of support, A, in (5.5). We take this region to be rectangular in the transformed space of x_i (in fact, the shape is unimportant) and so it is specified by two opposite vertices. Provided the component is initialized with at least some responsibility for each of the data points, it is straightforward to see that the maximum likelihood solution will be such that A is the minimal region that contains all of the points. Furthermore, this value will ensure that in subsequent EM steps the component continues to have non-zero responsibility for each point and therefore maintains this parameter value. In practice, then, we can set the parameter directly from the data and update only the mixing component $\pi_{\mathcal{C}'}$.

The remaining components form a mixture of Gaussians. EM update rules for this model are given in section 2.6. We omit, of course, the update of the covariances as they are known in advance.

The update rules for parameter estimates at the nth step are thus

$$\begin{aligned} r^n_{m,i} &= \frac{\pi^{n-1}_m \mathsf{P}_{\theta^{n-1}_m}(x_i)}{\sum_l \pi^{n-1}_l \mathsf{P}_{\theta^{n-1}_l}(x_i)} \quad ; \quad m = \emptyset, \varphi, 1 \ldots M \\ \pi^n_m &= \frac{\sum_i r^n_{m,i}}{|\mathcal{X}|} \quad ; \quad m = \emptyset, \varphi, 1 \ldots M \\ \mu^n_m &= \frac{\sum_i r^n_{m,i} x_i}{\sum_i r^n_{m,i}} \quad ; \quad m = 1 \ldots M \end{aligned} \tag{5.12}$$

They are iterated until convergence.

It is guaranteed that this procedure will converge to a local maximum of the model likelihood. However, the identity of that maximum is crucially dependent on the initial parameter values used to seed the optimization. EM shares this dependence with other hill-climbing approaches, whether first or second order. We can avoid it by using a Relaxation Expectation–Maximization (REM) technique as described in chapter 3. In this simple case REM yields an algorithm very similar to the simple deterministic annealing example treated by Rose *et al.* (1990). The differences are primarily in the presence of the mixing probabilities and the single non-Gaussian component.

The REM update rules differ only in the update of the responsibilities, which become, for a relaxation parameter β,

$$r^n_{m,i} = \frac{\pi^{n-1}_m (\mathsf{P}_{\theta^{n-1}_m}(x_i))^\beta}{\sum_l \pi^{n-1}_l (\mathsf{P}_{\theta^{n-1}_l}(x_i))^\beta} \tag{5.13}$$

(we have given the E-step according to the REM-2 algorithm; see section 3.5). The parameter β is increased gradually from near 0 to 1, with the EM iterations being run to convergence at each value of β. An extensive discussion of the properties of this algorithm is given in chapter 3

The number of cells

In the absence of simultaneous high-power microscopy, we generally do not know how many foreground cells are to be expected in an extracellular recording. As a result, this quantity must be estimated from the data along with the parameters of the spike waveform distributions. In the mixture model framework this is equivalent to determining the correct number of components.

As was pointed out in section 2.7.3, this is essentially a model selection problem. We have already examined at some length in sections 1.3 and 2.7.3 techniques appropriate to carrying out this selection. The use of the REM algorithm for learning makes available a particularly efficient and effective framework within which to apply these techniques, which we have called cascading model selection. This was discussed in section 3.6.

For the most part these techniques, described in part I of this dissertation, can be applied without modification. Two components of the mixture, the noise model $\mathsf{P}_\emptyset(\cdot)$ and the overlap model $\mathsf{P}_\varphi(\cdot)$ are always assumed to be present; thus, the model selection chooses between models with three or

more components.

5.9 Spike Shape Variability

The simple mixture model assumes that the action potential currents in each foreground cell are the same each time the cell fires, so that the only variability in the foreground spike waveform is due to the superposition of background spikes. In fact, this is rarely true.

Biophysically, one can imagine many reasons why the currents flowing across the somatic membrane might be variable. The concentrations of ions inside or outside the cell may vary. Ligand gated channels (for example, calcium-dependent potassium channels) may open on the membrane. A varying fraction, not large enough to prevent an action potential, of the sodium channels may be inactivated. Many of these conditions well depend on the recent activity of the cell, and this dependence will be examined more closely later. For the present, we will simply treat it as random variation.

5.9.1 Ratio methods

Some authors have argued (Rebrik *et al.* 1998; Zhang *et al.* 1997; Rinberg *et al.* 1999) that although the underlying action potential shape changes under these conditions, the *ratios* of the spike waveforms on the different channels should remain almost constant (disturbed only by the additive background noise). These ratios may be between maximal spike amplitudes, or between the magnitudes of the Fourier coefficients in various frequency bands. Such arguments are based on the same model as the ICA-based algorithms described earlier. The spikes recorded on the different channels are taken to be due to currents at a single point source which have been filtered differently by the extracellular medium through which they passed and by the electrode tip. If the source waveform (in the Fourier domain) is $S(\omega)$ the recorded signal on the nth channel will be $R_n(\omega) = F_n(\omega)S(\omega)$ where F_n is some linear filter. As the source changes, then, the spike shapes also change; but by taking the ratio of the recorded spike shapes $R_n(\omega)/R_m(\omega) = F_n(\omega)/F_m(\omega)$ we divide out the source signal and obtain a stable measure.

Once again, the arguments advanced against the applicability of ICA-models in, at least, neocortical tissue, apply here. The most severe is the fact that the simple model of one-source-multiple-detectors does not hold in preparations where the action potential travels over significant sections of cell membrane. In neocortical and hippocampal pyramidal cells, for example, action potentials are known to propagate over the dendrite (Stuart and Sakmann 1994; Stuart *et al.* 1997) and different electrode tips will record spikes due to different parts of the membrane (Buzsaki and Kandel 1998). In discussions of spike variability a further difficulty presents itself. The spread of the action potential across the membrane is known to be variable, depending on the recent firing activity of the cell

(Spruston *et al.* 1995; Svoboda *et al.* 1997). Thus, not only are the sources recorded by the different electrode tips spatially distinct, but these sources can vary in a distinct manner. As a result, there is reason to expect ratio methods to be inadequate in such preparations.

5.9.2 Models of the variability

Unable to remove the intrinsic variability in the waveforms, we seek to model it. In this section will we discuss models in which the underlying spike shapes are independent and identically distributed. Following this treatment, in section 5.10, we will discuss models which capture the dependence of the spike shape on the recent firing history of the cell.

Unconstrained Gaussians

One approach, attractive for its mathematical simplicity, is to model the underlying spike shape variability as Gaussian. If this model were correct, each observed spike waveform from a given cell would be the sum of two Gaussian random variates, and thus, would itself be Gaussian distributed. We have no independent data source from which to establish an appropriate covariance matrix for the intrinsic variability, and so the covariance must be learned along with the mean spike waveform. The measured background covariance can only provide a lower bound.

The general EM iterations for the arbitrary Gaussian mixture are as in (5.12), with the addition of a re-estimation rule for the mth covariance matrix

$$\Sigma_m^n = \frac{\sum_i r_{m,i}^n (x_i - \mu_m^n)(x_i - \mu_m^n)^T}{\sum_i r_{m,i}^n} \tag{5.14}$$

If the background covariance has been whitened, we can enforce the lower bound set by the background by diagonalizing the Σ_m^n obtained in this way, resetting any eigenvalues less than unity to 1, and then rotating back into the original space. If V is the matrix of eigenvectors of Σ_m^n, and the binary operator $\max(\cdot\,,\cdot)$ is taken to act element by element

$$\Sigma_m^n \leftarrow V \max(V^\mathsf{T} \Sigma_m^n V, I) V^\mathsf{T} \tag{5.15}$$

In the case of the background process, the superposed nature of the signal led us to expect it to be approximately Gaussian. In contrast, we have no reason to believe that the intrinsic variability should give rise to a Gaussian process, and so the validity of this model will rest entirely on the experimental evidence. In practice, cell waveform distributions in the macaque data set seemed to be well approximated in this fashion only if they did not fire bursts of closely spaced action potentials. The case of the bursting cells will be discussed more thoroughly below.

One issue introduced by the use of unconstrained Gaussians is the multiplicity of parameters. In

a D dimensional space, each component of the simple Gaussian model contributes only D parameters to the model. In contrast, the unconstrained Gaussian contributes $D(D+1)/2+D$ parameters. As the number of parameters increase the dangers of over-fitting and of being trapped in local maxima increase. The REM algorithm can alleviate the second of these to some extent, however strategies to reduce the complexity of the model are useful. On approach is to constrain the number of non-unit eigenvalues (in the background-whitened space) in each model. This leads (in the unwhitened space) to a mixture model, analogous to the mixture of factor analyzers model of Ghahramani and Hinton (1996). We will not explore this any further here, turning instead to a non-Gaussian generalization.

Hierarchical Gaussian mixtures

As was pointed out above, there is no *a priori* reason to expect the intrinsic variability to be Gaussian distributed. While such a model may provide a successful approximation in certain examples, it is insufficient to account for all of the observed data. Therefore, we will now investigate a non-parametric alternative.

The mixture model, which we have taken as the basic statistical model underlying probabilistic cluster analysis, has another rôle in the statistical literature. A mixture of relatively simple components (such as Gaussians) is often used to approximate a more complicated density, about which little is known *a priori*. Such an approach is called "non-parametric" because there is no explicit generative model of the density. It is not suggested that the data are in fact generated by any sort of mixture process. Rather, the mixture model is being used as an extremely flexible substrate for density approximation. (Compare the use of radial basis function networks in the function approximation literature).

Our alternative, then, is to fit an **hierarchical mixture model** in which the generative distribution for each cell is itself a mixture. We shall employ a mixture of Gaussians, each with a covariance matrix equal to that of the measured background noise. In a sense, this approximation may be viewed as identifying a small handful of "canonical" spike shapes, which span the range of possibilities. The generative process selects one of these shapes and then adds background noise to produce the observed spike waveform. In fact, the intrinsic waveform of the spike (before addition of the background) is not discrete in this fashion. This problem is mitigated by the fact that the Gaussian density provides significant probability mass in the region in between the selected points. We may think of the model as "tiling" the true density with a small set of identically shaped ellipses, the shape being set by the background covariance.

Let us write down the density that results from such a model. Suppose there are M clusters, with mixing proportions π_m. Each cluster is modeled by a mixture of P Gaussians, with mixing proportions $\rho_{m,p}$, means $\mu_{m,p}$ and unit covariances (we assume that we have whitened the background

process). The parameter set for the model is $\theta = \{\pi_m\} \cup \{\rho_{m,p}\} \cup \{\mu_{m,p}\}$. We have,

$$\mathsf{P}_\theta(\mathcal{X}) = \sum_i \sum_m \pi_m \sum_p \rho_{m,p} (2\pi)^{-d/2} e^{-\frac{1}{2}\|x_i - \mu_{m,p}\|^2} \tag{5.16}$$

If we distribute the factor π_m into the sum over p and write $\psi_{m,p} = \pi_m \rho_{m,p}$ it becomes clear that this density is identical to that derived from a mixture of $R = M \times P$ Gaussians. Indeed, any hierarchical mixture in which the total number of Gaussians is R, even if there are unequal numbers of components used to describe each cell, will yield the same form of the density.

This poses a serious problem from the point of view of model selection. Conventional model selection procedures may indicate the correct density from among a group of candidates. But, how are we to decide which components belong to which cell? Probabilistically, any such assignment would be equally valid, including the "flat" option in which every component represents a single cell. In short, from a probabilistic point of view, there is no such thing as a hierarchical mixture!

We may choose to exploit additional information in order to group the Gaussians.

One approach is as follows. Begin by fitting a mixture of a large number of Gaussians (all with unit variance) to the data. The actual number is not of great importance, provided it is significantly larger than the number of cells expected. It may be chosen arbitrarily, or by a model selection method. Then, form a graph, with one node for each Gaussian. An edge between two Gaussians is included if the densities exhibit a significant degree of overlap, that is, if the distance between their means is smaller than some chosen threshold. Each of the connected subgraphs that results is taken to represent a single cell. Such an approach would be similar in spirit, although different in detail, to that proposed by Fee *et al.* (1996a) (a detailed discussion of the relationship to their method is outlined in section section 5.14).

Alternatively, the additional information might be encoded as a prior on the parameters within a group. For example, we might expect that the means of the components that describe a single cell will lie close together, and will themselves be drawn from a Gaussian density of small variance.

In both these approaches, one or more control parameters must be chosen arbitrarily: either the overlap threshold for the graph formation, or the form and extent of the prior. In many cases, these parameters may be chosen anywhere within a fairly broad range of values, with identical results. However, it is in the case when the waveforms from two or more cells are very similar, and where the model selection procedure is thus most important, that the results become most sensitive to the choice of parameters.

In section 5.10.2 we will introduce a third approach to the resolution of the ambiguity in the hierarchical mixture likelihood, suitable for modeling variability intrinsic to bursts of action potentials. There, a dynamic model is proposed, in which the components representing a single cell are tied together by a learnt Markov transition structure. In that view, components belong to the same

cell provided that the timing of spikes that fall within them is consistent with a simple burst model.

5.10 Dynamic Models

In the models discussed thus far each spike waveform is generated independently of all others. We turn now to models in which the latent variables are dependent on each other.

5.10.1 Refractory period

One simple feature of the firing process has not yet been accounted for in any of our models. This is the occurrence of the refractory period, a short period after each action potential during which the cell that fired will not fire again. As it stands, the mixture model has no representation of the time of any event. We will discuss shortly a model in which time is explicitly represented. For the moment, though, it is possible to account for the refractory period by a simple modification to the basic mixture model. The method presented in the following may be applied to any of the various mixture models we discussed above; for simplicity we shall develop it in the case of the simple Gaussian mixture of section 5.8.

The joint data log-likelihood for such a model was given in section 2.6

$$\ell_{\mathcal{X},\mathcal{Z}}(\theta) = \sum_i \sum_m z_{m,i} \left(\log \pi_m - \frac{1}{2} \log |2\pi\Sigma_m| - \frac{1}{2}(x_i - \mu_m)^T \Sigma_m^{-1} (x_i - \mu_m) \right) \tag{2.17}$$

In the refractory case this expression remains valid for most data and parameter values; the exception is provided by sequences of $z_{m,i}$ that violate the refractory constraint by assigning to the same cell events that fall within a refractory period of each other, for which the log-likelihood diverges to $-\infty$. In taking the expected value of the log-likelihood, however, the probability of such a sequence is 0, and so we can discount this possibility. The expected log-likelihood under the distribution $\mathsf{P}_{\theta^{n-1}}(\mathcal{Z} \mid \mathcal{X})$ retains the general mixture form of (2.8)

$$\begin{aligned} Q^n(\theta) &= \sum_i \sum_m \mathcal{E}_{z_{m,i}|x_i,\theta^{n-1}}[z_{m,i}] \log \pi_m \mathsf{P}_{\theta_m}(x_i) \\ &= \sum_i \sum_m s^n_{m,i} \left(\log \pi_m - \frac{1}{2} \log |2\pi\Sigma_m| - \frac{1}{2}(x_i - \mu_m)^T \Sigma_m^{-1} (x_i - \mu_m) \right) \end{aligned} \tag{5.17}$$

except that, as we will see below, the expected values of the $z_{m,i}$ are different from before. To remind ourselves of this difference we use the notation $s^n_{m,i}$ for these new responsibilities, reserving the symbols $r^n_{m,i}$ for the responsibilities in the non-refractory case.

To obtain the new responsibilities, consider first the simple case where only two spikes have been observed and the second appears within a refractory period of the first. We have a joint distribution

over $z_{m,1}$ and $z_{m',2}$ with

$$\mathsf{P}\left(z_{m,1}, z_{m',2}\right) = \begin{cases} 0 & \text{if } m = m' \\ r^n_{m,1} r^n_{m',2}/Z & \text{otherwise} \end{cases} \tag{5.18}$$

where $Z = \sum_m \sum_{m' \neq m} r^n_{m,1} r^n_{m',2}$ is an appropriate normalizing constant. The expected values we seek are then just the marginals of this joint distribution, for example,

$$s^n_{m,1} = \sum_{m' \neq m} r^n_{m,1} r^n_{m',2}/Z = r^n_{m,1}(1 - r^n_{m,2})/Z \tag{5.19}$$

where we have used the fact that $\sum r^n_{m',i} = 1$.

This result easily generalizes to the case of many spikes

$$s^n_{m,i} = \frac{r^n_{m,i}}{Z_i} \prod_{i,j \text{ refractory}} (1 - r^n_{m,j}) \tag{5.20}$$

where Z_i is the appropriate normalizer and the product is taken over all spikes that are fall within one refractory period (before or after) the i spike.

The M-step is still a weighted Gaussian estimation as before, the weights now being the new responsibilities $s^n_{m,i}$.

5.10.2 Sparse hidden Markov models

Bursts

The intrinsic variability of spike waveforms is not entirely random for all cells. Many pyramidal cells, both in neocortex and in the hippocampus, sometimes fire action potentials in bursts. Action potentials within a burst are closely spaced (as little as 1ms apart), and the cell does not have enough time to recover from one before the next begins. Thus, the membrane currents associated with later action potentials are likely to be smaller, and a smaller portion of the dendritic membrane will participate in such spikes. As a result, the spike waveforms recorded later in the burst may be quite different from those associated with isolated action potentials.

In this section we will construct a statistical model to describe the change in action potential during a burst. At first glance, one might think that a sufficient model would have the expected spike waveform depend on the immediately preceding interval. In fact, the situation is considerably more complex than this. For example, the third spike in a regular burst will usually be smaller than the second, even though the preceding interval is the same. At the same time, it is true that after a longer interval the cell has had more time to recover and so the spike waveform is closer to the normal case.

Faced with the complexity of the mechanisms underlying the change in spike waveform during a burst, we will not attempt a biophysical model. Instead, we will use a simple statistical model that will capture the variation empirically.

A statistical model

The statistical model that we consider is a constrained version of the Hidden Markov Model (HMM). Each cell is modeled by a single HMM, which is independent of all of the others. In practice, it is often useful to use HMMs to model only a subset of the cells in a recording — those that exhibit bursts — and use Gaussians or other static distributions to describe the others.

The output symbols of the underlying Markov model are either complete spike waveforms represented as vectors (the events of the previous discussion) or a zero vector. The vast majority of symbols in any string generated from the Markov model will, in fact, be zero and so these models are sparse in the sense of chapter 4. The observed vector is the sum of the Markov model output and a random vector drawn from the background process. Thus one may think of the output distributions of the states of the HMM as Gaussians, centred either on zero or on a mean waveform which is to be learned. The output density is thus identical to that of the hierarchical Gaussian mixture model discussed in section 5.9.2. The difference is that events are not chosen from this density independently. This change in the model provides another approach to breaking the ambiguity inherent in the hierarchical model.

A Markov model describes a discrete time process. We choose to discretize time in fairly large steps, usually 0.5ms. The measured output symbol for any given time-bin is a spike waveform if the identified time (that is, the peak or centre of mass) of some event falls within that bin. Otherwise, the output symbol is taken to be 0.

The transition matrix of the Markov model is constrained so as to embody the structure expected from a bursting cell. This constrained structure is sketched in the left-hand part of figure 5.10. Each of the grey circles in this figure represents a state of the HMM. The left column of states all have zero output symbol and represent the cell in a non-firing state. States in the right column represent firing events in the cell and have non-zero output distributions. These distributions are indicated on the stylized event feature plot to the right. Each state is associated with a Gaussian output distribution indicated by an elliptical boundary. Together, these distributions "tile" one of the elongated clusters in the data set.

Each heavy arrow in the HMM diagram represents an allowed transition: where there is no arrow the transition probability is set to 0 and remains at this value throughout the learning process. The states are arranged in a "ladder" with states lower down the ladder corresponding to greater recent firing (and therefore greater inactivation of channels). The upper left-hand state is the "ground" state, in which the cell will be found after a long period of inactivity. Only two transitions are

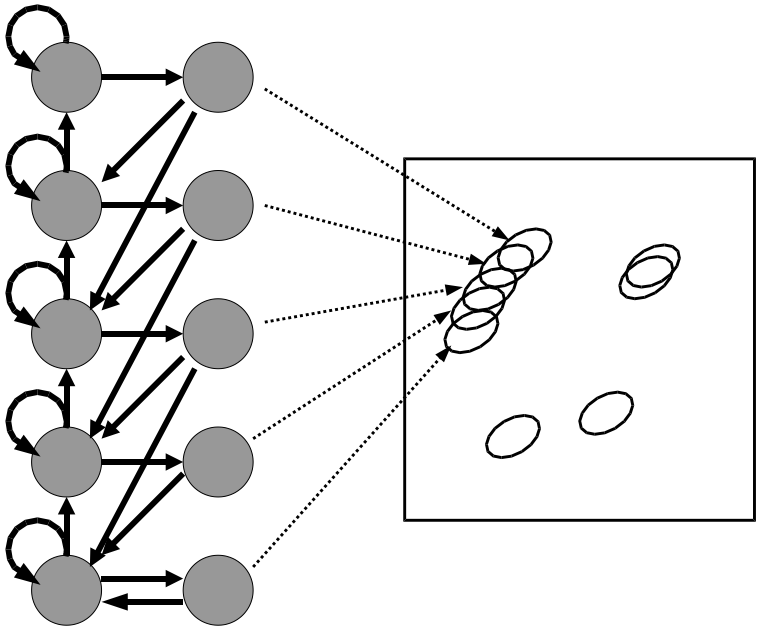

Figure 5.10: The HMM transition structure

possible from this state: the cell either fires an action potential, making the transition to the state on the right, or else remains in the same state. Once in the firing state, the cell makes a transition to a non-firing state below the ground state, thus preserving the memory of the recent firing. From this state, the cell can fire again, with a different output distribution, in which case it subsequently moves further down the ladder of states; it can remain in the same state; or it can make a transition up the ladder. This basic pattern is repeated for each of the rungs of the ladder.

Some features of this structure are worth pointing out. The only way for the cell to transition down the ladder is to fire. Once it fires it must enter a non-firing state and so cannot spike in successive time-bins; for 0.5ms bins this effectively enforces a short refractory period. If the cell finds itself some distance down the ladder, but does not subsequently fire for a number of time-steps, it will relax back to the ground state with an exponential decay profile.

Learning with HMMs

A learning algorithm for mixtures of sparse HMMs was discussed in section 4.4. Sparse HMMs were defined in that section to produce two types of output: either a null symbol, $\emptyset$, or a numerical value. When considering mixtures of sparse HMMs we introduced a third type of output, the symbol ϕ, which was detected when two or more of the component HMMs emitted non-null outputs in the same time-step.

In the current application an output is defined for each 0.5ms time-bin as follows. If no event

has its peak (or centre of mass) within the bin the observation is taken to be ø. In most cases, if an event does peak within the bin, the observation is the reduced vector representation of that event. However, if the event has been classed as an outlier, then the symbol ¢ is observed. Outlier events are identified in three ways during our procedure. First, the waveform may exhibit a double peak or other heuristically excluded property during event extraction. Second, the event may fall outside the principal subspace during dimensionality reduction. Finally, it may be assigned with high probability to the outlier mixture component. This last poses a problem, since we cannot know before fitting is complete which events are to be classified in this way; but we also cannot fit the mixture of HMMs accurately without knowing which observations are collisions. In practice, this circularity is resolved by dynamically marking as a collision any event that is assigned to the outlier cluster with a probability that exceeds some set threshold on a given iteration.

Given these definitions, the learning algorithms of section 4.4 can be employed to optimize the mixture parameters.

5.11 Mixed Models

There is no reason to expect that all of the foreground cells present in a particular recording will exhibit the same type or degree of variability. A single site may yield some cells that tend to fire in bursts of action potentials; some that fire isolated, but stochastically variable spikes; and some that exhibit no detectable intrinsic variability at all. Thus, it is often useful to be able to combine the three types of waveform model we have discussed in this chapter — the fixed covariance Gaussian of the simple mixture model; the mixture of Gaussians of the hierarchical mixture model; and the sparse hidden Markov model — in a single overall mixture.

The framework in which to do so is provided by the mixture of sparse hidden Markov models discussed above, and at greater length in section 4.4. In particular, we observe that both the single, fixed covariance Gaussian and the mixture of fixed covariance Gaussians may both be expressed as special cases of the sparse HMM, with transition matrices constrained differently from the "ladder" of figure 5.10.

The simple fixed-covariance Gaussian model is equivalent to a two-state HMM. One state (say, the first) has null output, the other has an output distribution given by the Gaussian model. To reproduce the basic model exactly, the columns of the transition matrix must be identical. The augmented transition matrix (including the initial state probabilities; see section 4.1.1) is of the form

$$T_m = \begin{pmatrix} 0 & 0 & 0 \\ 1-\rho_m & 1-\rho_m & 1-\rho_m \\ \rho_m & \rho_m & \rho_m \end{pmatrix} \tag{5.21}$$

Here ρ_m represents the firing probability per time-step associated with the mth model of the overall mixture. It is related to the mixing probability π_m as follows. Suppose the total number of events in the training data (with collisions counted twice) is N and the total number of HMM time-steps is T. Given the stationarity assumption of the mixture, we expect there to be $\pi_m N$ spikes from the mth cell in this data, and so the probability of a spike per time-step is $\rho_m = \pi_m N/T$.

The transition matrix given in (5.21), allows for the cell to fire in adjacent time-bins with probability ρ^2. In fact, it is convenient to exploit the HMM transition structure to enforce a refractory period without requiring the scheme of section 5.10.1. In section 5.10.2 we achieved this by requiring that the model return to a null state after firing. For 0.5ms time-steps, this enforced a short, but reasonable refractory period. Thus, we alter the transition matrix to

$$T_m = \begin{pmatrix} 0 & 0 & 0 \\ 1-\rho_m & 1-\rho_m & 1 \\ \rho_m & \rho_m & 0 \end{pmatrix} \tag{5.22}$$

The value of the firing probability ρ_m must now be corrected. The new relationship is $\rho_m = \pi_m N/(T - \pi_m N)$.

The mixture of Gaussians model for a single cell is implemented similarly. For a P component mixture the HMM now contains $P+1$ states, one with null output (again, we take this to be first) and the others with output distributions corresponding to the components of the mixture. If the mixing probabilities of the cell model are $\pi_{p,m}$ and the overall mixing probability of this cell model within the hierarchical mixture is π_m we define densities by $\rho_{p,m} = \pi_{p,m}\pi_m N/(T - \pi_{p,m}\pi_m N)$. We write $\rho_m = \sum_p \rho_{p,m}$. Then the augmented transition matrix, corrected to enforce a refractory period, is given by

$$T_m = \begin{pmatrix} 0 & 0 & 0 & \cdots & 0 \\ 1-\rho_m & 1-\rho_m & 1 & \cdots & 1 \\ \rho_{1,m} & \rho_{1,m} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \rho_{P,m} & \rho_{P,m} & 0 & \cdots & 0 \end{pmatrix} \tag{5.23}$$

Having converted each non-Markov model into a sparse hidden Markov model whose transition matrix embodies the appropriate structure, we can then proceed to learn the parameters using the algorithm described in section 4.4. In general, learning in such a model is more computationally expensive than in the basic mixture models. Thus, if no cells in a given data set appear to fire in bursts, so that the ladder-structure HMMs will not be needed, it is preferable to use the mixture model directly, possibly with the refractory modification of section 5.10.1. However, once the parameters are learned, the corresponding SHMMs can be constructed by the procedure given in this

section. These SHMMs can then be used for on-line spike recognition, as described in section 5.13.

5.12 On-line Learning

In many applications of spike sorting, recognition must be carried out in close to real time. In scientific experiments, for example, feedback in the form of sensory stimulus changes or even neural stimulation might need to be triggered within milliseconds of a particular pattern of action potentials being recorded. In neural prosthetic applications, neural activity needs to be transformed into a "motor" action on a similar time scale.

For the most part, such demands constrain the inference, or spike recognition, stage of sorting (to be discussed below) rather than the learning. We may collect an initial segment of data without the real time demands, train on these data off-line and then perform on-line inference.

However, it is useful to update the parameter estimates as more data are collected. For one thing, these updates will refine the estimates, yielding progressively more reliable data. As a result, it might be possible reduce the length of the initial training segment, leading to a smaller training down-time prior to on-line recognition.

More important, though, is the fact that in almost all recording situations, the parameters are likely to drift over time. Such drift generally occurs due to minute changes in the relative positions of the cells and electrodes, thus changing the recorded spike waveforms. Even without such physical displacement, however, the statistics of spiking of the different cells, which enter into the models in the form of mixing parameters or transition probabilities in the HMM, may change. For example, cells may switch between more or less bursty modes of firing in association with varying levels of drowsiness (or anesthesia) in the subject.

In this section we discuss techniques for on-line parameter adaptation. Similar techniques will allow both refinement of the estimates as new data come in, as well as tracking of slow drift in the parameters. We discuss these techniques as though the parameters are to be updated each time a new spike is observed. In practice this level of immediacy is unnecessary, and it is more efficient to collect spikes for a short period (say 1s) and apply the updates in a batch form.

5.12.1 Incremental EM

We showed in section 1.8 (following Neal and Hinton 1998) that the free energy interpretation of EM can be used to justify some variants on the basic algorithm. One of these is an incremental version in which the parameters are updated one data point at a time. This approach is valid in cases where both the observations x_i and the latent variables y_i are independent and drawn from fixed distributions, and so the conditional distribution $\mathsf{P}_\theta(\mathcal{Y} \mid \mathcal{X})$ factorizes over the y_i. Of the models we have discussed here, this is true only of the mixtures.

The iterations for the incremental EM algorithm, in the notation of section 1.8, are as follows.

IE-step: Choose some i. Maximize $F_i(p_i, \theta^{n-1})$ and leave the remaining $p_j, j \neq i$ unchanged.

$$\begin{aligned} p_i^n(y_i) &= \mathsf{P}_{n-1}(y_i \mid x_i) \\ p_j^n(y_j) &= p_j^{n-1}(y_j) \end{aligned} \tag{5.24}$$

M-step: Maximize F with respect to θ holding p constant.

For a mixture model, the probability distribution $p_i^n(y_i)$ is simply the set of responsibilities $r_{i,m}^n$, $m = 1 \ldots M$ and the M-step involves maximizing the weighted log-likelihood $\sum_i r_{i,m}^n \mathsf{P}_{\theta_m}(x_i)$ for each component.

The on-line version of this algorithm is different only in that there is no choice of i. The data are simply handled, one by one, as they arrive from an unlimited stream. The M-step update only involves, of course, the data collected to this point. We shall assume that the initial parameter values chosen are very close to the true values, being the result of training on a separate, off-line, data set. This assumption means that even though data are not revisited, the responsibilities assigned to them remain reasonably valid. An alternative approach is outlined in the next section.

Fortunately, for Gaussian mixtures (and indeed many other mixture models) it is not necessary to store all of the past responsibilities and observations in order to update the parameters in the M-step. We derive the M-step update rule for a general mixture of unconstrained Gaussians; the result for the various constrained Gaussian models used for spike sorting will follow immediately.

The usual M-step updates for a Gaussian mixture, given N data points, are

$$\pi_m^n = \frac{\sum_{i=1}^N r_{m,i}^n}{N} \tag{5.25}$$

$$\mu_m^n = \frac{\sum_{i=1}^N r_{m,i}^n x_i}{\sum_{i=1}^N r_{m,i}^n} \tag{5.26}$$

$$\Sigma_m^n = \frac{\sum_{i=1}^N r_{m,i}^n (x_i - \mu_m^n)(x_i - \mu_m^n)^T}{\sum_{i=1}^N r_{m,i}^n} \tag{5.27}$$

The $(N+1)$th data point, x_* arrives, triggering the $(n+1)$th update of the parameters. We calculate the responsibilities, r_{m*} of each of the components for this point in the usual fashion. According to the incremental EM algorithm, then, the new estimate for π_m is

$$\pi_m^{n+1} = \frac{1}{N+1} \sum_{i=1}^{N+1} r_{m,i}^{n+1} = \frac{1}{N+1} \left(\sum_{i=1}^{N} r_{m,i}^n + r_{m*} \right) = \frac{N}{N+1} \pi_m^n + \frac{1}{N+1} r_{m*} \tag{5.28}$$

where we have used the fact that $r_{m,i}^{n+1} = r_{m,i}^n$ for all $i < N+1$. Similarly, we find that (writing

$R_m^n = \sum_{i=1}^N r_{m,i}^n = N\pi_m^n$)

$$\mu_m^{n+1} = \frac{1}{R_m^{n+1}} \sum_{i=1}^{N+1} r_{m,i}^{n+1} x_i = \frac{1}{R_m^{n+1}} \left(\sum_{i=1}^{N} r_{m,i}^n x_i + r_{m,*} x_* \right) = \frac{R_m^n}{R_m^{n+1}} \mu_m^n + \frac{1}{R_m^{n+1}} r_{m*} x_* \tag{5.29}$$

Finally, the corresponding result for Σ_m^{n+1} follows by rewriting (5.27) as

$$\Sigma_m^n = \frac{\sum_{i=1}^N r_{m,i}^n x_i x_i^T}{\sum_{i=1}^N r_{m,i}^n} - \mu_m^n \mu_m^{n\,T} \tag{5.30}$$

from which we find that

$$\Sigma_m^{n+1} = \frac{R_m^n}{R_m^{n+1}} \left(\Sigma_m^n + \mu_m^n \mu_m^{n\,T} \right) + \frac{1}{R_m^{n+1}} r_{m*}\, x_*\, x_*^T - \mu_m^{n+1} \mu_m^{n+1\,T} \tag{5.31}$$

5.12.2 Parameter adaptation

When the update algorithms described above are used in an on-line fashion (without revisiting any data), the impact of each succeeding point on the parameter estimates grows progressively smaller. If the parameters are varying slowly, this is an unfortunate state of affairs, since information about the new values will be incorporated at an ever decreasing pace. Indeed, even if the parameters are stable, but the initial estimate of the model was far from the true value, this state of affairs is not too promising. The reason (stated here in terms of the incremental EM algorithm for mixtures, although it applies equally to the HMM) is that the responsibilities that were calculated for the first few data points become increasingly inaccurate as the model is optimized. While the effect of these early values on the estimate is diluted by ever more incoming data, leading to the correct result in the limit, convergence would be more rapid if we had a mechanism to "forget" them. (Note that the incremental EM algorithm as described by Neal and Hinton (1998) avoids this problem by revisiting all the data with some probability).

Notice that each of the update rules derived in the previous section (5.28), (5.29), (5.31) has the form of a weighted sum of old information and new. The form of amnesia we seek can be achieved by the simple measure of adjusting the weights in this sum to favour the new data.

One approach is suggested by Nowlan (1991). In this view, the optimal parameter values are maintained by a group of sufficient statistics; for the mixture of Gaussians, these statistics are $R_m^n = \sum_i r_{m,i}^n$, $S_m^n = \sum_i r_{m,i}^n x_i$ and $SS_m^n = \sum_i r_{m,i}^n x_i x_i^T$. Knowing the values of these statistics at any iteration n we can calculate the parameter values $\pi_m^n = R_m^n / \sum_m R_m^n$, $\mu_m^n = S_m^n / R_m^n$ and $\Sigma_m^n = SS_m^n / R_m^n - \mu_m^n \mu_m^{n\,T}$. The update rules derived in the previous section can then be easily expressed in terms of these sufficient statistics

$$R_m^{n+1} = R_m^n + r_{m*}\,; \qquad S_m^{n+1} = S_m^n + r_{m*}\, x_*\,; \qquad SS_m^{n+1} = SS_m^n + r_{m*}\, x_*\, x_*^T \tag{5.32}$$

The proposal made by Nowlan (1991) introduces a factor $\gamma < 1$ to regulate the decay of older information. The sufficient statistic update rules are replaced with these:

$$R_m^{n+1} = \gamma R_m^n + r_{m*} \; ; \qquad S_m^{n+1} = \gamma S_m^n + r_{m*}\, x_* \; ; \qquad SS_m^{n+1} = \gamma SS_m^n + r_{m*}\, x_*\, x_*^T \tag{5.33}$$

We can thus derive the parameter update rules under this approach. If we write N_e^n for $\gamma \sum_m R_m^n$ we obtain,

$$\pi_m^{n+1} = \frac{R_m^{n+1}}{\sum_m R_m^{n+1}} = \frac{\gamma R_m^n + r_{m*}}{\sum_m (\gamma R_m^n + r_{m*})} = \frac{N_e^n \pi_m^n + r_{m*}}{N_e^n + 1} \tag{5.34}$$

and

$$\mu_m^{n+1} = \frac{S_m^{n+1}}{R_m^{n+1}} = \frac{\gamma R_m^n \mu_m^n + r_{m*}\, x_*}{R_m^{n+1}} = \frac{N_e^n \pi_m \mu_m^n + r_{m*}\, x_*}{R_m^{n+1}} \tag{5.35}$$

with a similar result for the covariance update. Comparison with (5.28) and (5.29) suggests that the term N_e^n plays the rôle of an effective number of data. Note that $N_e^{n+1} = \gamma N_e^n + 1$. Thus if $N_e^n = (1-\gamma)^{-1}$ then $N_e^{n+1} = N_e^n$ and otherwise $N_e^{n+1} > N_e^n$. The effective number of data climbs until it reaches the value $(1-\gamma)^{-1}$ and then remains constant. Thus we may think of this approach as limiting the effective number of data used.

Such an approach is seen to be reasonable in situations where the parameters change at a rate linked to the number of data measured (or in the case where such adaptation is needed to speed on-line convergence given poor initial parameter values). In the spike sorting example, however, we expect the parameter variation to occur at a rate constant in time, even if the overall spike rate varies. We would like the effective number N_e^n to be dependent on the recent firing rate of the cells being recorded.

The formulation in terms of an effective number of data makes this easy. We replace the term N_e^n in the above by a firing-rate dependent term that varies in time $N_e(t)$. The dependency on firing rate might set $N_e(t)$ to the number of spikes recorded within a window. It should be borne in mind that this approach is different to simply using only the last $N_e(t)$ data points to estimate the parameter values. The estimates are based on all previous data; however, the estimate derived from these data is weighted as though it was derived from only $N_e(t)$ points.

5.12.3 Limited look-ahead forward–backward

The scheme described in the previous section is appropriate for on-line adaptation of the parameters of mixture models, whether of the simple Gaussian type, or more elaborate. What about the dynamic hidden Markov model, proposed in section 5.10.2? At first glance, the situation appears impossible. Recall that to perform even a single E-step of the learning algorithm requires a traversal through all of the data by the forward–backward algorithm. It would seem, then, that we cannot even begin to learn the parameters of the model until all of the data have been collected.

Of course, this is not exactly true. If the parameters were stationary we would expect that parameter estimates derived from a moderately long sequence of data would be reasonable, and affected only marginally by the incorporation of additional observations. The critical point is that the influence of later observations on earlier state and transition estimates is diminished by mixing in the Markov chain. Thus, although in principal the backward pass of the inference algorithm should begin at the very end of the data set, if it is instead begun earlier, only the immediately preceding state estimates (those within one mixing time) will be substantially incorrect. This feature is exploited by Boyen and Koller (1999) in the context of general dynamic probabilistic networks. For the sparse hidden Markov model the situation is further improved, because, as was argued in section 4.3.2, long stretches of null observations tend to "reset" the model. "Long," in this context, refers to the mixing time of the null-state restricted Markov chain; in the spike sorting context this is the time taken for a cell to reset after a burst and thus may well be on the order of 20ms.

The incremental approach to learning the HMM thus involves re-running the backward pass of the forward–backward algorithm only as far back as the last segment of moderate silence. To be conservative, one might discount state estimates in the M-step until they become "protected" by a stretch of nulls, although in practice this rarely makes any difference. In any case, if one realigns the notion of the "current" time to the last estimate that can be trusted, we may think of this procedure as taking into account a short sequence of data in the future. Thus the name **limited look-ahead forward–backward algorithm**.

As new state information becomes available it is combined with the earlier information by a procedure analogous to (5.29) and (5.31), with the state estimates $s^n_{p,m,i}$ replacing the responsibilities. The update of the transition matrix is similar in spirit to (5.28), but differs slightly. We write $t_{pq,m,*}$ for the new transition estimate and $S^n_{q,m} = \sum_{i=0}^{N-1} s^n_{q,m,i}$ to obtain

$$T^{n+1}_{pq,m} = \frac{\sum_{i=1}^{N+1} t^{n+1}_{pq,m,i}}{\sum_{i=0}^{N} s^{n+1}_{q,m,i}} = \frac{\sum_{i=1}^{N} t^n_{pq,m,i} + t_{pq,m*}}{S^{n+1}_{q,m}} = \frac{S^n_{q,m}}{S^{n+1}_{q,m}} T^n_{pq,m} + \frac{1}{S^{n+1}_{q,m}} t_{pq,m*}. \tag{5.36}$$

For non-stationary parameters we can implement adaptive rules by weighting the updates by an effective data size just as in (5.33) and following. In this case, since a new estimate is generated at every time-step whether a spike occurred or not, we do not need to worry about varying the effective number of data, and we simply choose a fixed value of the decay constant γ.

5.13 Spike Time Detection

Given the model structure and parameters, the third and final stage of the spike sorting process is the inference of the firing times. To perform this inference accurately, and in particular to resolve overlapped spikes, we will return to the full superposition model (5.2), using the distributions for

the firing indicators $c_{m,\tau}$ and waveforms $S_{m,\tau}$ derived from the learnt mixture model. Many, if not most, previous spike sorting approaches have not made this distinction: inference is performed on extracted events using a cluster assignment model and is not actively distinguished from the learning of the model. Such an approach leaves three issues unresolved. First, the threshold-based event detection heuristic of section 5.5 can be improved upon once the true spike shapes have been determined. Second, if all events are to be clustered, the sorting process must occur off-line, ruling out experiments in which rapid feedback about the cells' responses is needed. Third, the clustering procedure has discarded the superposed events, or else collected them into an unresolved overlap cluster, rather than resolving them into their constituent spike forms.

The correct solution to the inference problem involves a search through all possible combinations of spike arrival times, and is computationally prohibitive. Lewicki (1994) suggests that with optimized programming techniques, and suitable, but severe approximations, it is possible to complete this search in close to real time on a computer workstation. We shall not review his implementation here; the interested reader is referred to the cited paper. Instead, we discuss an alternative set of approximations that lead to a straightforward, single-pass, greedy algorithm. This approach is particularly well-suited to parallel implementation on arrays of digital signal processors (DSPs).

We shall derive the procedure in the context of the sparse hidden Markov models of section 5.10.2, where the output distribution of each component is either null or a Gaussian of fixed covariance (set by the background). As was seen in section 5.11, other cell models that we have considered can also be expressed in this form, and so the detection method we discuss will apply equally well to the simple Gaussian model of section 5.8 or to the hierarchical Gaussian mixture of section 5.9.2. It will not, however, apply to the unconstrained Gaussian model of section 5.9.2 without considerable modification.

The basic structure of the scheme is as follows. At each time-step we begin by estimating the prior probability distribution over the states of each SHMM, based on our estimates of the states at the preceding time-step. Using these probabilities, and the data recorded around the given point in time, we obtain the occupancy likelihoods for each of the firing states of each of the models, along with the likelihood that no spike was observed. We accept the event associated with the largest likelihood. If this optimal likelihood is for no spike, then we re-derive the posterior state distribution for each model as though a null symbol was observed. If, on the other hand, the optimal likelihood is due to one of the firing states, we assume that the appropriate model is, in fact, to be found in that state. The corresponding mean spike waveform is subtracted from the recorded data; and again the likelihoods of the remaining models having fired, or of there having been no second spike are calculated. This is repeated until no more spikes remain to be accounted for at this time-step. The initial state probabilities for the next step are then inferred by transitions from the posterior estimates of the states at the current time.

This is a recursive procedure similar to the forward step of the coupled forward–backward algorithm. We will examine in detail a single step of the procedure in analogy to the treatment of section 4.4.2.

We assume that at the $(i-1)$th time-step, the current state probability estimates are given by $E_{p,m,i-1}$[6]. Since the Markov transitions are taken to be independent, these are propagated forward to provide initial estimates of the probabilities at the ith step by the relation

$$\tilde{E}_{p,m,i} = T_m E_{p,m,i-1} \tag{5.37}$$

We need to assess the probability of a spike being present on this time-step. However, we are no longer dealing with pre-extracted and aligned spike waveforms and so the spike, if any, may have occurred at any point within the time interval under study. We can measure the probability by the maximal output of a simple matched filter. Suppose that the pth component of the mth model has a non-null output distribution, with mean waveform (transformed into the time domain from whatever subspace was used to fit) given by $S_{p,m}(t)$. We assume that the background has been whitened, so that the covariance of this output distribution, and all the others, is I. The joint log-likelihood of a spike having been generated from this particular component (that is, that the state variable $y_{m,i} = p$) at a particular time τ, under the observed trace $V(t)$, is

$$\begin{aligned}
&\log \mathsf{P}\left(V(t) \mid y_{m,i} = p, \tau\right) \\
&\quad \propto \quad -\frac{1}{2}\int dt\ \left(V(t) - S_{p,m}(t-\tau)\right)^2 \\
&\quad = \quad \int dt\, V(t) S_{p,m}(t-\tau) - \frac{1}{2}\int dt\, V(t)^2 - \frac{1}{2}\int dt\, S_{p,m}(t-\tau)^2
\end{aligned} \tag{5.38}$$

while the likelihood that there was no spike is simply

$$\log \mathsf{P}\left(V(t) \mid \emptyset\right) \propto -\frac{1}{2}\int dt\, V(t)^2 \tag{5.39}$$

The spike time τ will be assumed to lie within the short interval under consideration for this time-step. The integrals over t extend through all time; although we will soon drop the integral of $V(t)^2$, and the others can be limited to the support of $S_{p,m}(t-\tau)$. Note that the final term in (5.38) is, in fact, independent of the spike time τ; we will therefore write $\alpha_{p,m} = \int dt\, S_{p,m}(t)^2$ for the total power in the waveform associated with the distribution (p,m).

We can combine these expressions with our prior expectations of each state given by $\tilde{E}_{p,m,i}$, and drop the common term that depends only on $V(t)$ to obtain the following weighted matched-filter

[6] We adopt the same conventions for subscripts as we did in section 4.4, so that p refers to the state, m to the model and $i-1$ to the time-step.

outputs:

$$\mathcal{F}_{p,m,i}(\tau) = \int dt\, V(t) S_{p,m}(t-\tau) - \frac{1}{2}\alpha_{p,m} + \log \tilde{E}_{p,m,i}/\delta \tag{5.40}$$

$$\mathcal{F}_{\emptyset,i}(\tau) = \log \sum_{\emptyset_{p,m}=1} \tilde{E}_{p,m,i}/\delta \tag{5.41}$$

where δ is the length of the time-step. The first of these is calculated only for non-null states, while the sum in the second is over all null states. Up to a shared constant term, these two expressions indicate the posterior probabilities of a spike having occurred at time τ from component (p,m) (5.40) and of no spike having occurred (5.41), respectively. The first of these may be seen to be result of a matched filter with impulse response $S_{p,m}(-\tau)$ being applied to the data.

It is here that we make our greedy step. We select the single largest probability from among the values (5.40) and (5.41), over all times τ within the time-step window (in fact, if this maximum lies at the boundary of the interval we extend the search to the closest peak in the filter value). If this is $\mathcal{F}_{\emptyset,i}$ we assume no spike occurred in the interval. In this case the new state estimates are given by

$$E_{p,m,i} = \emptyset_{p,m} \frac{\tilde{E}_{p,m,i}}{\sum_p \emptyset_{p,m} \tilde{E}_{p,m,i}} \tag{5.42}$$

in agreement with (4.43).

If, however, the maximum is achieved by one of the filter outputs, say $\mathcal{F}_{p^*,m^*,i}(\tau^*)$, we assume that the corresponding spike really did occur. In this case we set $E_{p^*,m^*,i}$ to 1 and all other state probabilities for the m^*th model to 0. We then subtract from the data stream the waveform $S_{p,m}(t-\tau^*)$ and recalculate the filter outputs to see if perhaps another spike occurred as well. In practice, since the filters are linear, we can actually subtract the appropriate filtered version of the waveform directly from the filter output. The procedure is then repeated, with the m^*th model discounted. We continue to subtract and repeat until no further spikes are detected.

The procedure described here yields reasonable results in many cases. In the context of non-trivial HMM transition matrices, however, it can be improved upon by the use of the standard Viterbi decoding algorithm of HMM theory, adapted in a manner similar to the coupled forward–backward algorithm discussed in section 4.4. In particular, we note that the forward pass of the decoding does not need to be run to completion before the backward pass (in which the most probable states are identified) can begin. Instead, the optimal sequence can be determined each time a block of nulls of sufficient length is encountered (see section 4.3.2).

5.14 Comparison with Previous Work

Spike sorting is by no means a new problem. Extracellular recording has been a routine electrophysiological method for decades, and single units have been isolated from voltage traces for many years. Nonetheless, it is only quite recently, as multiple electrode recording has become more widespread and as fast computers have become easily available, that interest in fully automatic spike sorting has arisen, and a full statistical analysis of the problem has not, to date, been carried out.

In this section, we review some previous approaches, both manual and automatic, used or proposed for spike sorting. The discussion of prior art has been postponed to this late stage because it is now, armed with the full statistical analysis of the problem, that it will be possible to properly understand the techniques proposed and their shortcomings, if any. We shall find that most approaches to be discussed will address only a subset of the issues brought out in our treatment.

This review of earlier work does not purport to be exhaustive. As might be expected of a subject so fundamental to experimental neuroscience, hundreds of papers have been published on spike sorting. The few that are mentioned below have been selected on two bases: first, they are the best examples of the different common classes of algorithm; and second, in many cases they have been quite influential in the creation of the current work. In some cases, mention of earlier work has already been made in the course of the development above, in which case only a note to that effect will appear here.

5.14.1 Window discriminators

The most basic tool for the detection of spikes in extra-cellular recording is a simple threshold device known as a Schmidt trigger. In the last few decades a slightly more sophisticated version of this venerable tool has come into use, known as the **window discriminator**, and it is this that we shall describe here. The discriminator is usually a hardware device — although the same functionality can easily be implemented on a computer — designed to identify spikes from a single cell. The amplified signal from the electrode is compared to a manually-fixed threshold applied to either the signal voltage or to its derivative. Each time the threshold is triggered, the subsequent waveform is displayed on an oscilloscope (or computer) screen. Observing these waveforms, the user sets a number of time-voltage windows that bracket the waveforms that he wishes to identify as foreground spikes. Any triggered waveform that passes through all of these windows is accepted as a spike, and the time of occurrence is logged.

These devices have typically been used in conjunction with manual isolation of a single spike, so that all that needs to be done with the windows is to distinguish this single waveform from the background. However, software versions of the same device may allow multiple sets of windows to bracket spikes of different shapes (or more than one hardware discriminator may be used on the

same signal), and in some cases spikes from more than one cell can be reasonably detected in this manner.

We can view this procedure as a special case of the manual clustering approach to be described below. The trigger simultaneously extracts and aligns the waveforms. As can be seen from figure 5.5C, as long as the threshold crossing is detected in the analogue signal (that is, there is no, or else only extremely fast, sampling involved) this procedure yields reasonably well-aligned spikes; alignment to a centre of mass is, however, very slightly better. The time-positions of the windows relative to the threshold crossing select the dimensions of the waveform space used to cluster, and the voltage-extents of the windows set the cluster boundaries within this space. Thus, the clustering is constrained to occur within an axis-aligned subspace and the cluster boundaries are constrained to be rectangular. One advantage to this scheme over many standard clustering packages is that it allows the user to select the appropriate dimensions from among all of the axial directions. Another advantage (in terms of manual clustering) is that the high-dimensional space of waveforms is compactly visualized on a two-dimensional screen. Nonetheless, the restrictions on subspace dimensions and on cluster shape can be quite restrictive.

5.14.2 Manual clustering

The advent of multi-wire electrodes, and the availability of commercial software, has popularized the use of clustering approaches to spike sorting. The basic framework of these approaches is as follows. Event waveforms are extracted using a fairly basic threshold trigger. In general, no attempt is made to resample or to realign the event. These waveforms are then grouped into clusters, sometimes by an *ad hoc* clustering algorithm, but often by having the operator draw out the cluster boundaries in various two-dimensional projections. There is no separate spike-detection phase; membership of the clusters, along with the recorded time of threshold crossing, fully specifies the estimated spike identity and time. Examples of procedures of this sort have been described by Abeles and Goldstein (1974), Gray *et al.* (1995), Rebrik *et al.* (1998) and many others.

In general, the clustering is carried out in a subspace of reduced dimension. Above, we pointed out that window discriminators can be viewed as selecting a subset of event coordinates for clustering. Other techniques that have been employed are those that were described in section 5.7.2; hand-picked features, often derived from the spike waveform in a non-linear fashion, are common (see, for example, products from DataWave Technologies), while PCA has also been used (Abeles and Goldstein 1974; Gray *et al.* 1995). In section 5.9.1 we also discussed some proposals to reduce dimensionality in such a way as to suppress spike-shape variability.

Frequently, the cluster shapes are constrained to be rectangular; we pointed out above that this is implicit in the window discrimination approach to clustering, while in many explicit clustering packages it appears to be imposed as a matter of programming convenience. Other computer pack-

ages allow elliptical (for example, the latest product from DataWave Technologies) or more general polygonal (such as the program xclust, written by M. Wilson) boundaries.

In detail, these techniques can certainly be improved in the light of the analysis that has appeared here. Event alignment, discussed in section 5.7.1, would reduce the apparent cluster noise; projection into the noise-whitened robust principal component space, discussed in section 5.7.2, would improve separation. On the issue of the quality of the resultant clustering, however, we expect that the human eye is a sufficiently sophisticated pattern recognition engine to yield fairly accurate results, provided that it is assisted by a proper presentation of the data. One of the advantages to this approach is that it obviates the need to find explicit general models of the spike-shape variability. The operator can, instead, assess the pattern of variability on a cell-by-cell basis. (Of course, clustering packages which restrict the cluster boundaries to be rectangular can hamper this flexibility.)

The difficulties in such methods fall into four groups. First, if the cluster assignments provide the final estimates of spike identity there is no way to resolve overlapped waveforms. Second, the lack of a probabilistic underpinning reduces the degree to which the quality of the solution can be assessed. With probabilistic methods the likelihood of the optimal fit can provide some indication of whether the data have been reasonably modeled or not. Furthermore, a probabilistic technique leads to "soft" or "fuzzy" clusters, which, in turn, lend themselves to the assessment of the degree of confidence with which any given assignment can be made. Both of these features are lacking the "hard" clustering schemes that are commonly used. The third set of issues arises from the fact of human intervention. Spike assignments generated in this fashion may be not be reproducible across different experimenters. Further, the need for considerable experimenter input limits the degree to which the method can be scaled. As we acquire the technology to record from hundreds of electrodes at once, the need for an operator to examine waveforms from each one becomes a prohibitive obstacle. Finally, clustering schemes such as these cannot operate on-line in real time. Thus, they are inappropriate for experiments in which immediate feedback is needed, nor can they be used in neural prosthetic applications.

5.14.3 Automatic techniques

Gaussian models

Lewicki (1994) provides an analysis of the problem that is closest in spirit to that provided here. The model described is based on a single spike waveform per cell, with added spherical Gaussian noise. While the algorithms are derived from an explicitly Bayesian point of view, the resulting steps are similar to those that we describe in section 5.8. Many of the details, however, are different. Thus, Lewicki treats the alignment of the waveform within the sampled event as a latent variable and re-estimates its value on each fitting iteration, while we attempt to eliminate the variation in alignment

by the technique described in section 5.7.1. His model contains no explicit outlier component, and instead low occupancy models need to be inspected and possibly rejected by the operator.

A significant difference lies in his approach to the model selection problem. Rather than the cascading model selection procedure that we have proposed, which might be viewed as a form of divisive clustering, he initially fits a mixture with more components than expected and then fuses adjacent clusters together based on the calculation of an approximate Bayes factor.

The most significant shortcoming in Lewicki's proposal is the lack of more sophisticated models for the spike distribution from a single cell. We described in section 5.9 the reasons that we might expect a single Gaussian to be an inadequate model. Similar concerns led Fee *et al.* (1996a) (see below) to abandon the explicitly probabilistic approach. The methods described in this dissertation demonstrate that more powerful models capable of modeling the intrinsic variability in the spike waveforms, can, indeed, be implemented within the probabilistic point of view, thereby gaining all of the advantages implied by that approach.

Agglomerative clustering

In response to Lewicki (1994), Fee *et al.* (1996a) argue, as we did in section 5.9, that in many cases the distribution of waveforms from a single cell does not appear to be Gaussian. They therefore propose an agglomerative clustering scheme which is *ad hoc* in the sense of not being probabilistically founded. The scheme is as follows.

Events are extracted and aligned to a centre of mass calculated in a manner similar, though not identical, to (5.7). The resultant vectors are first partitioned into small clusters by a "recursive bisection" algorithm somewhat similar to divisive k-means. These clusters are then agglomerated into larger groups. Two clusters are grouped together if they exhibit a large "boundary interaction"; that is, roughly, if the density of points in the region of the boundary between them exceeds some threshold.

This may be viewed as an *ad hoc* version of the hierarchical mixture model described in section 5.9.2. The hierarchical mixture provides all the advantages, described above, of the "soft" probabilistic approach. Furthermore, the agglomeration procedure proposed in section 5.9.2 is more satisfying in that it requires explicit overlap of the components. This is made possible by the use of a mixture model, in which the component densities are able to overlap, rather than k-means clustering in which the clusters are compelled to be disjoint.

ART networks

Another proposal that has appeared in the literature is the use of a generic neural network classifier. Oghalai *et al.* (1994) suggest the application of an ART-2 network (the acronym ART comes from the adaptive resonance theory of Carpenter and Grossberg 1987a, 1987b, 1990). This is a neural

network architecture designed for unsupervised clustering problems, and as such appears to be a likely candidate. Closer inspection, however, reveals some weaknesses. In particular, ART implies an odd distance metric in which clusters whose centers have smaller L_1 norms are favoured. Furthermore, as each incoming vector is classified, the center is updated by taking the point-by-point minimum of the old center and the new point. Neither of these details seems to match the noise characteristics we have seen. ART is also a sequential clustering scheme, in which the order in which the data are presented is important. Moore (1989) has argued that it is particularly sensitive to noise in the data. Overall it cannot be thought of as any better than any of the *ad hoc* clustering schemes discussed in section 2.1.

5.14.4 Spike time detection

Some authors have made the same distinction between clustering and spike time detection that we have. In general, they have been motivated by a desire to correctly identify overlapped spikes within the recording, although these techniques may often bring with them the additional benefits that we described in section 5.13.

Lewicki (1994) proposes that the space of all possible waveform overlaps can be searched by the introduction of some approximations and the use of efficient programming techniques. It should be noted that in making this claim, he is addressing detection in the context of a Gaussian clustering model that yields a single mean waveform for each cell. For the more complex distributions, involving multiple components for each cell, the computational difficulty is further increased. Nonetheless, in situations where adequate computational power is available, this is an attractive approach. However, the greedy approximation made in section 5.13 is expected to exhibit slightly improved scaling.

Roberts and Hartline (1975) (see also Roberts 1979) propose an "optimal" linear filtering algorithm, similar to the standard Wiener matched-filter. Expressed in the frequency domain, the filter used to detect the mth spike shape is given by the transform of the associated waveform divided by the sum of the power in the other waveforms and the noise. This filter has the property of responding minimally to the other waveforms (and to noise), while maintaining its output in response to the target waveform at a fixed level. In essence, the filters transform the data to a basis in which the different spike shapes are orthogonal; in this basis overlaps are easily identified.

In the context of the tetrode recordings described here, this approach has not proven to be very successful. The problem seems to be that spike shapes from different cells are spectrally similar enough that the attempted orthogonalization is impossible. The matched filtering technique described in section 5.13 differs from this one in that no effort is made to orthogonalize the targets. Instead, the interaction between the filters is handled explicitly by subtracting the waveform with the largest response from the data and re-filtering. While slower, this approach yields more reliable results.

It should be noted that Gozani and Miller (1994) report success with this technique. Their recordings were made with multiple hook electrodes arranged along a nerve bundle. Spike waveforms might have differed in their propagation velocity along this nerve, a feature which would have facilitated orthogonalization. For cortical tetrode data, or other data recorded within neuropil with a multi-tip electrode, differences in propagation velocity are quite unlikely to be detected.