Decoding

Balazs B Ujfalussy

9/10/2017

This is a demo of decoding stimulus orientation from neuronal activity in the visual cortex. The data is from Ecker et al., *Decorrelated Neuronal Firing in Cortical Microcircuits*, Science, 327, 584 (2010). http://bethgelab.org/datasets/v1gratings/

The RData file contains a list containing 8 preselected cells (>1 Hz firing rates, clear spikes) from a single experiment in given day. The fields of the list contain the following:

date date and time stamp when the session was recorded subject identifies the monkey used in the session conditions specifies the orientation and contrast used, selected for > 0.05 contamination contamination of the single units tetrode specifies the tetrode a single unit was recorded on; for tetrode grid layout, see supplementary material of Ecker et al. (2010) spikes contains binned spikes single units x conditions x time bins x repetitions times times aligned to bin centers

The experimental data is loaded from the `Data/Ecker/` folder - data_v1_binned_static_ses2.RData. First we load the data and see what it contains using the summary command:

```
load("Data/Ecker/data_v1_binned_static_ses2.RData")
summary(data)
```

```
##               Length Class  Mode
## date               1 -none- character
## subject            1 -none- character
## conditions        32 -none- list
## contamination      8 -none- numeric
## tetrode            8 -none- numeric
## spikes        276480 -none- numeric
## times             90 -none- numeric
```

The spikes field contains an array with the spikes recorded: cells x conditions x time bins x repetitions.

```
print(dim(data$spikes), include=FALSE)
```

```
## [1]  8 16 90 24
```

The are two variables for condition: orientation and contrast. We have 8 orientations and 2 contrast levels. Now we will focus on the high contrast stimuli.

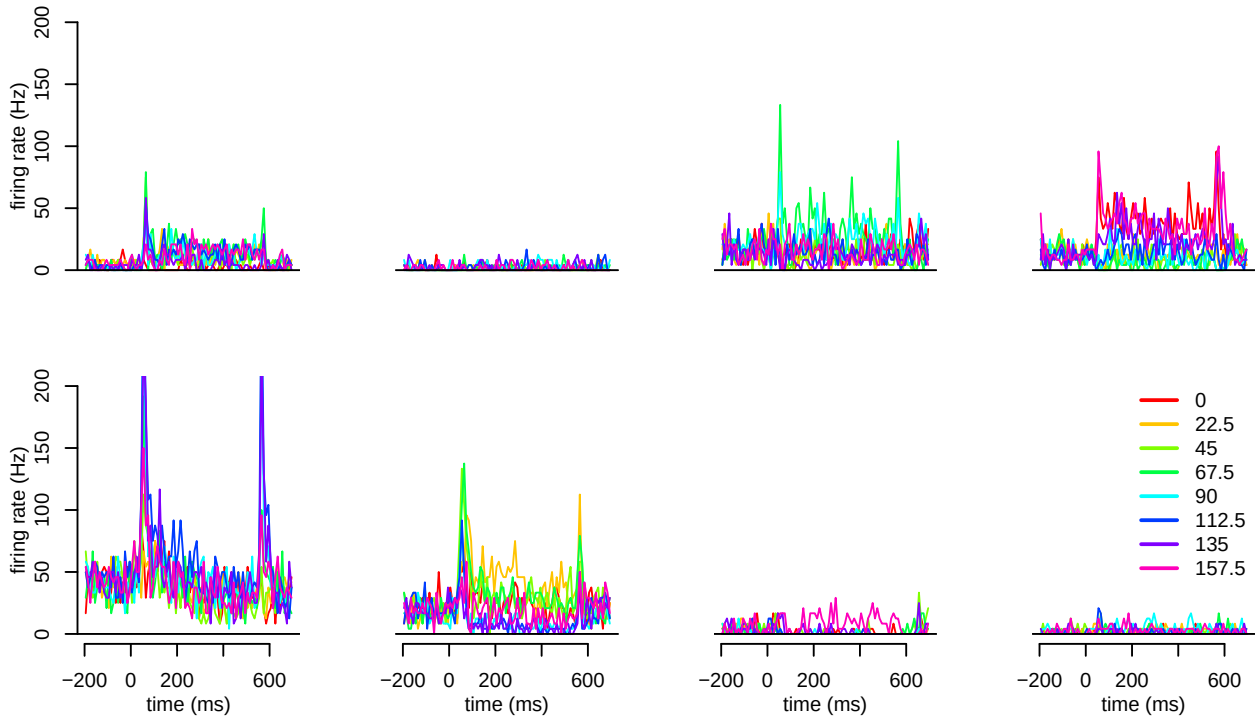
Data for decoding

First, we extract contrast and orientation variables for plotting, and the firing rate - approximated as the average spike count - for each time point:

```
high.contrast <- data$conditions['contrast',] == 10
data$spikes <- data$spikes[,high.contrast,,]
data$conditions <- unlist(data$conditions['orientation',high.contrast])
oris <- unlist(data$conditions)
```

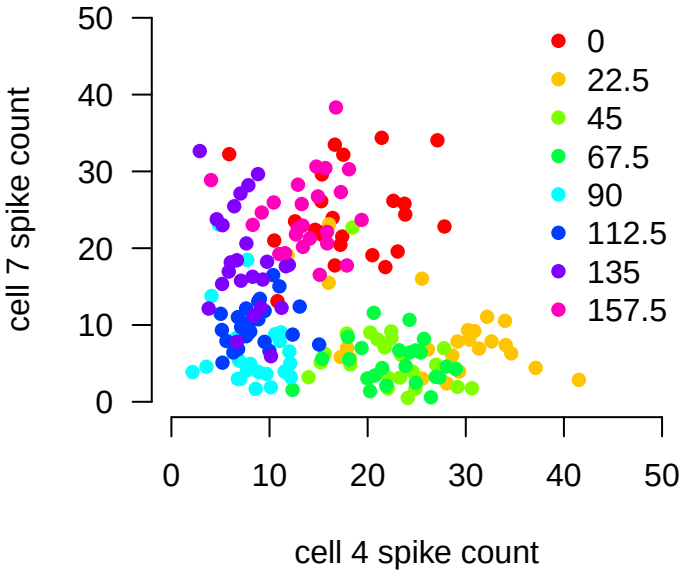
```
n.cells <- dim(data$spikes)[1] # number of cells
n.conditions <- dim(data$spikes)[2] # number of conditions
L <- dim(data$spikes)[3] # length of recordings
par(mfcol=c(2,4)); par(mar=c(3,3,1,1)) # plotting subfigures
for (i.cell in 1:n.cells){
    rates <- matrix(0, L, n.conditions) # this is going to be the rate matrix for the cells
    for (i.condition in 1:n.conditions){
        sp <- data$spikes[i.cell,i.condition,,]
        rates[,i.condition] <- apply(sp, 1, mean)*100 # Hz
    }
    matplot(data$times, rates, t="l", col=rainbow(180)[ceiling(oris)+1], lty=1, lwd=1, axes=F, ylim=c(0
    abline(h=0)
    if ((i.cell %% 2) == 0){
      axis(1); mtext('time (ms)', 1, 2, cex=0.7)
    }
    if (i.cell < 3){
      axis(2); mtext('firing rate (Hz)', 2, 2, cex=0.7)
    }
    # readline(i.cell)
}

legend('topright', leg=unique(oris), col=rainbow(180)[unique(ceiling(oris))+1], lwd=2, lty=1, bty='n')
```



Then, we extract their spike counts on each trial - we will do decoding based on the total spike count.

We have 8 cells, 8 conditons and 24 repetitions. Remember, the vector *oris* contains the orientations. Now I select cell 4 and 7 which seems as an active cell showing nice stimulus-dependent responses, and plot their joint activity:

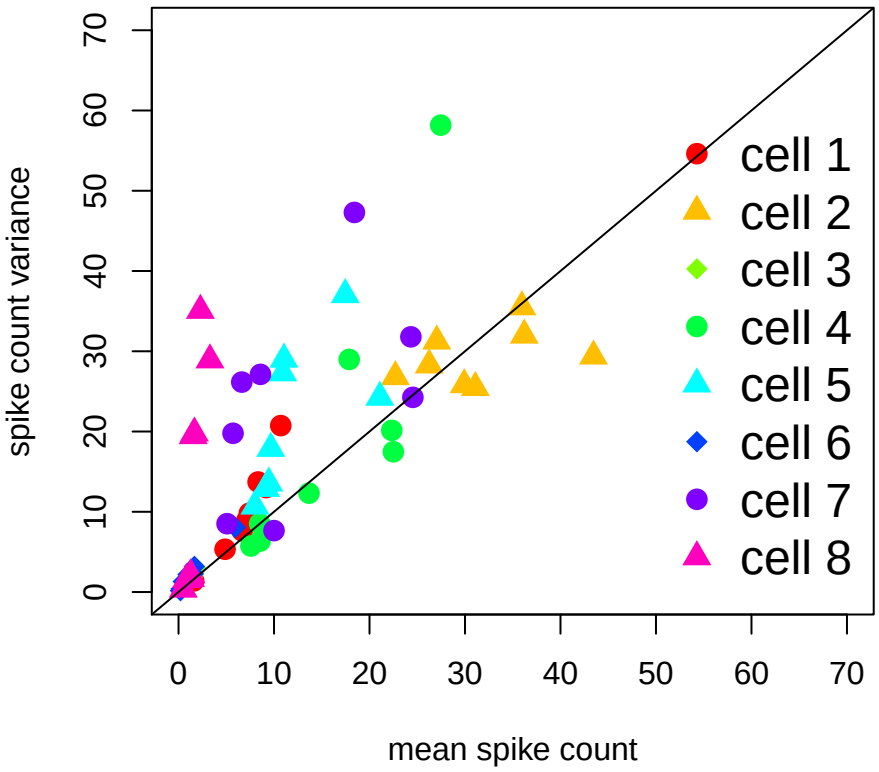


Bayesian decoding

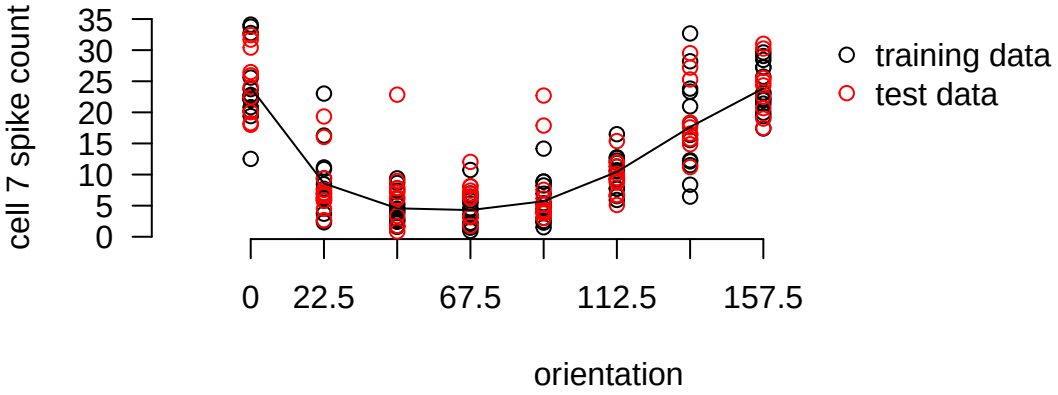
To do Bayesian decoding, we need a generative model for how the spike counts depend on the stimuli. Here we will call the stimulus s and the response of cell i as r_i. The model specifies the distribution of possible responses given the stimulus: $P(r_i|s)$. Here we will assume, that cells are independent, so $P(r_i, r_j|s) = P(r_i|s)P(r_j|s)$.

One possible candidate of the spike count model is the Poisson model, when the responses are generated from a Poisson distribution with the mean parameter depending on the stimulus: $P(r|s) = \text{Poisson}(\lambda(s)) = \lambda^r e^{-\lambda}/r!$. Importantly, the Poisson distribution has a single parameter, λ, the mean spike count. This means that it is very easy to estimate, but also, that it assumes that the mean and the variance of the distribution is tightly coupled!

To test whether for these cells the Poisson distribution is a good model for the spikes we will plot the variance of the spike counts in the function of the mean, for each cell separately:



The variance is typically higher than the mean - so the spike count distribution does not look like it was coming from a Poisson distribution, we conclude that the errors are not too large. We will now perform Bayesian decoding based on the activity of cell 7. We use 14 randomly selected trials for training and the rest 10 trials for test, and estimate λ as the mean spike count for the training trials of a given orientation.



Now, for any spike count in a given trial, our Bayesian decoder will tell us the posterior distribution over the stimuli, which is, according to Bayes's rule: $P(s|r) = P(r|s)P(s)/P(r)$.

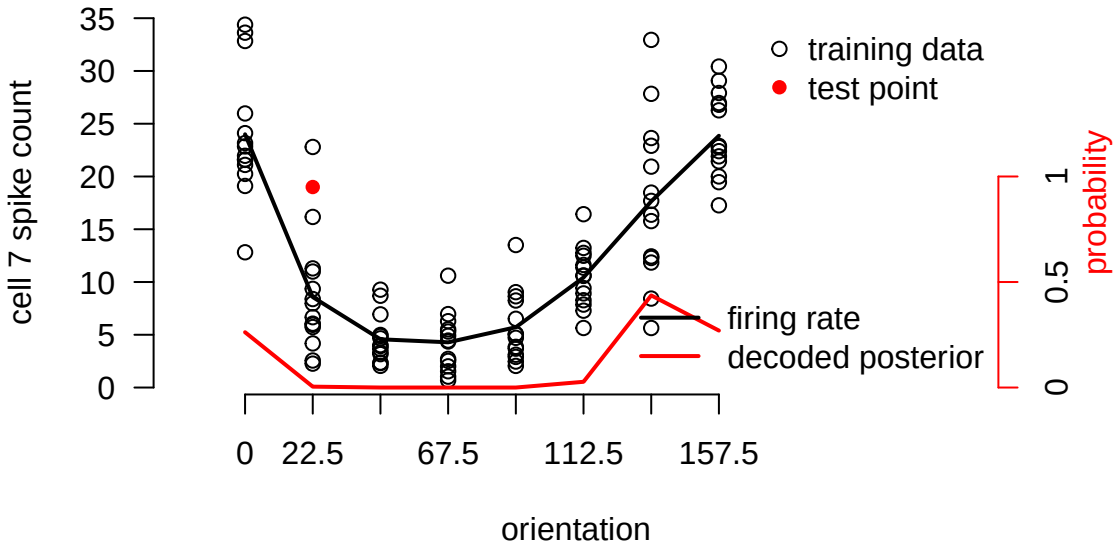
- $P(r|s) = \text{Poisson}(\lambda(s))$ is the Poisson distribution defined above. When we do decoding, we know the response, r, and we are interested in the stimulus r. So we need the evaluate $\text{Poisson}(r|\lambda(s))$ for all possible stimulus values.
- $P(s)$ is the prior over the stimulus. In this case all stimuli are equally likely, so it is uniform, and we will omit it.
- $P(r)$ is the marginal likelihood - the probability of observing this particular response irrespective of the actual stimulus. This is a constant (independent of the stimulus), so again, we don't need to deal with it when doing decoding.

So we end up having the simple equation: $P(s|r) \propto P(r|s)$, i.e., to decode the spike counts all you need to

evaluate the likelihood associated with a given orientation, and normalise them. This is done by the following function:

```
decode.Pois <- function(lambda, r){
  # lambda: stimulus dependent firing rate
  # r: observed spike count
  post <- dpois(r, lambda)
  normalised.post <- post / sum(post)
  normalised.post
}
```

Now, I illustrate Bayesian decoding in the case of the third trial for the second orientation, which contains 19 spikes.



There are a couple of important observation to make:

- The result of the decoding is not a single orientation.
- The result of the decoding is a probability distribution, shown by the red line. Several different orientations are consistent with the data, observing 19 spikes. The posterior quantifies exactly this: what is the probability that a particular orientation underlies the stimulus.
- The posterior has two peaks - we can not decide between two orientations, that look almost equally likely.
- If required, we can easily return an orientation: the most likely, of the one, that minimises certain error function.

Performance of the decoder

We will quantify the performance of the decoder (θ) in the following way:

- we apply it to each test data (responses, i.e., spike counts, r_i)
- we calculate the log of the probability of returning the correct orientation $\log P(s = s_i|r_i)$
- we add these logs for each datapoint to get the perfogmance: $\theta = \sum_i \log P(s = s_i|r_i)$

```
LL <- 0
for (i.ori in 1:8){
  for (i.trial in 1:N.test){
    post <- decode.Pois(lambda, test.7[i.ori,i.trial])
    LL <- LL + log(post[i.ori])
```

```
  }
}
```

So the performance of the decoder is θ =-160, which means that the probability that for all test data it will provide the right answer is e^{θ}. Now we evaluate it for an average stimulus. We had 80 test data altogether, so the average log likelihood was $\theta/80$ =-2, which is correct with probability 0.13. For 8 orientations, the chance decoder would be correct with probability $1/8 = 0.125$. Not too much, but we only used a single neuron.

Homework problem

Evaluate the performance of the decoder in the case of two neurons, e.g., neuron 4 and neuron 7.