Exercise H: Poisson and Posterior Predictives

Benedikt Ehinger

27. Januar 2017

1 Poisson

Neuronal firing rates are commonly modeled by poisson processes. We observed two spiketrains and binned the spiketrains into firing rates (spikes / s). We want to compare the rates of both neurons. We think of modeling them as poisson distributions with two rates and the respective difference, but we are not sure if poisson is really appropriate. The Fano factor [1] is defined as:

$\frac{sd(data)^2}{mean(data}$

In a poisson process, the variance should be equal to the mean, thus the fano factor should be 1. Four our data the fano factors are quite high with both close to 1.2 . We need to check our model afterwards to see if such a high fano factor could appear just by chance, or if our model is not adequate.

You can find the data in h_exercise.csv

The model is as follows (replace all %% with the appropriate code)

```
model <- "
data {
  int<lower=0> n_counts1;
  int<lower=0> n_counts2;
  int<lower=0> counts1 %%
  int<lower=0> counts2 %%
}
parameters {
  %% # Parameter definition of lambda1
  %% # Parameter definition of lambda2
}
transformed parameters {
}
model {
  //
  %% the likelihood function 'Poission(lambda)' for lambda1
  %% the likelihood function 'Poission(lambda)' for lambda2
}"
```

[1] https://en.wikipedia.org/wiki/Fano_factor

Evaluate the model (example code in R) using:

```r
library(rstan)
samples <- stan(model_code=model,
                data=list(counts1=counts1,
                          counts2=counts2,
                          n_counts1=length(counts1),
                          n_counts2=length(counts2)),
                iter=1000, chains=4)
```

Your tasks are the following:

1. Code the model and get the MCMC sample to run.

2. Make sure the MCMC chains did not not converge

3. Plot and interprete the posterior density. Calculate $\delta = \lambda_1 - \lambda_2$.

4. We introduced earlier that we are not sure whether the model adequately captures the data (in other words, whether a poisson process can explain our data). We want to do posterior predictive checks. That means, we want to sample (=randomly generate) new data, based on our posterior parameter distribution and the model. In order to do so We will add a new block to our stan code, that automatically samples new data for us [2].

```
'generated quantities{
  int<lower=0> counts1_pred[n_counts1];
  int<lower=0> counts2_pred[n_counts2];
  for(k in 1:n_counts1)
    counts1_pred[k] = poisson_rng(lambda1);
  for(k in 1:n_counts2)
    counts2_pred[k] = poisson_rng(lambda2);
  }'
```

In each sampling step, the generated quantities block is evaluated once. Thus you randomly sample from two poisson processes given your current estimate of lambda1/2 in this sample.

Now we start with a visual posterior predictive check. Generate a histogram/density of your observed data and of 10 iterations of the posterior predictive.

I used the following R-Code to extract the samples and get them in a tidy [3] dataset

[2] In principle you can do this afterwards with your own code, i.e. in matlab or R. Sometimes this is more convenient, especially if the model takes a long time to rerun. In many simpler cases the quantities block is a good place to generate this new data

[3] ftp://cran.r-project.org/pub/R/web/packages/tidyr/vignettes/tidy-data.html

```r
params = melt(rstan::extract(samples,c('counts1_pred','counts2_pred')),
              value.name=c('counts'))
```

Compare them visually, do you think the model generates data that look equivalent to the observed one?

5. We want to calculate the fano factor for each posterior predictive sample we made and compare it to our actual data.

   You can first calculate the fano factor for each mcmc-iteration and save it in fanoPosPred.

   Then you can calculate: $mean(fanoPosPred < fanoActual)$ you will receive a posterior predictive p-value that evaluates whether the observed fano factor is a probable result given our fitted-model. Interprete it! Does our model hold?