Multi-neuron spike trains - distances and information.

Conor Houghton

Computer Science, U Bristol

NETT Florence, March 2014

In 1817 Stendhal reportedly was overcome by the cultural richness of Florence he encountered when he first visited the Tuscan city. As he described in his book Naples and Florence: A Journey from Milan to Reggio:

> *As I emerged from the porch of Santa Croce, I was seized with a fierce palpitation of the heart (that same symptom which, in Berlin, is referred to as an attack of the nerves); the well-spring of life was dried up within me, and I walked in constant fear of falling to the ground.*

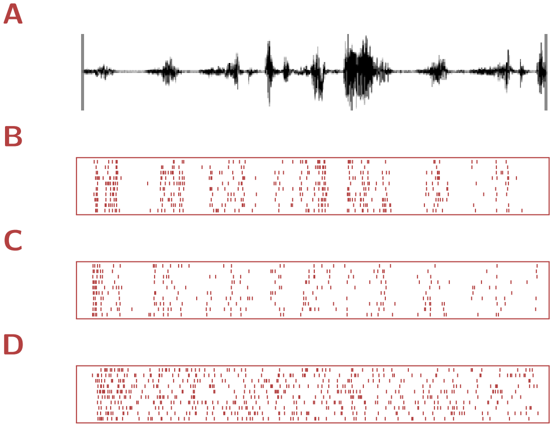
Wikipedia article `http://en.wikipedia.org/wiki/Stendhal`.

Overview.

- Mutual information for spaces with distances.
  - [Rederive a result due to Kraskov, Stöbauer and Grassberger (PRE 2004)]
- Spike trains and sets of spike trains as a space with distance.

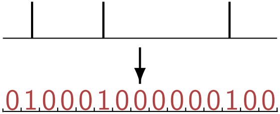
Spike trains.

Spiking responses in the auditory forebrain of zebra finch.



Classical approach I.

- Discretize.



- Split into words.

$$010001000000100 \to 01000, 10000, 00100$$

Bialek, de Ruyer van Steveninck, Strong and other coworkers, late 1990s.

Classical approach II.

- Estimate probability of words. For example, say $w_8 = 01000$ then estimate

$$p(w_8) \approx \frac{\#\ \text{occurrences of } w_8}{\#\ \text{words}}$$

- Calculate

$$H(W) = -\sum_i p(w_i) \log_2 p(w_i) = -\langle \log_2 p(w_i) \rangle$$

Bialek, de Ruyer van Steveninck, Strong and other coworkers, late 1990s.

Classical approach III.

- Conditional probability.

s_0	s_1	s_2	s_3	s_4	s_5
0 1 0 0 0	0 1 0 0 0	0 0 0 1 0	0 1 0 0 0	1 0 1 0 0	1 0 1 0 0
0 1 0 1 0	0 0 0 0 0	0 1 0 0 0	1 0 0 0 0	0 1 0 0 0	0 0 1 0 0
0 1 0 0 0	0 1 0 1 0	0 0 0 0 0	0 1 0 0 0	0 1 0 0 0	1 0 1 0 0
0 0 1 0 0	1 0 0 1 0	0 0 0 1 0	0 1 0 0 0	1 0 1 0 0	1 0 0 0 0
$H(W\|s_0)$	$H(W\|s_1)$	$H(W\|s_2)$	$H(W\|s_3)$	$H(W\|s_4)$	$H(W\|s_5)$

Bialek, de Ruyer van Steveninck, Strong and other coworkers, late 1990s.

Classical approach IV.

- Mutual information

$$H(W|S) = \langle H(W|s_i) \rangle$$

and

$$I(W;S) = H(W) - H(W|S)$$

Bialek, de Ruyter van Steveninck, Strong and other coworkers, late 1990s.

ms scale information in blow fly spike trains.

Bialek, de Ruyer van Steveninck, Strong and other coworkers, late 1990s.

Difficulties with the classical approach.

- Undersampling.
  - 100 ms words and 2 ms bins gives $2^{50} = 1125899906842624$ words.
  - Lots of clever approaches to this, for example Nemenman et al. (PRE 2004, BMC Neuroscience 2007) where a cunning prior is used for $p(w_i)$.
- Sampling bias.
  - An even distribution will never give equal counts for each word, giving different $p(w_i)$.
  - Lots of clever approaches to this too, see Panzeri et al. (J Neurophys. 2007).

Many fixes but still . . .

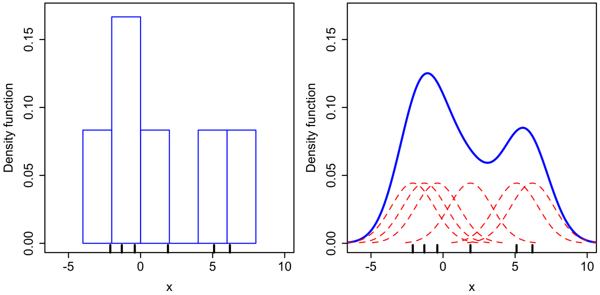
- Neuron - neuron mutual information.
- Maze - neuron mutual information.
- Mutual information with multiple units.

Approaches to probability estimation

- Parameteric approach.
  - See Gillespie and Houghton (JCN 2011).
  - . . . or Yu et al., (Front. in Comp. Neuro. 2010).
- Non-parametric approach
  - Histograms.
  - Kernel density estimation (KDE).
  - kth nearest neighbor (kNN).

We will see that KDE and kNN lead to more-or-less the same formula.

Kernel density estimation.



$$p(x) = \frac{1}{n}\sum_i k(x - x_i)$$

Picture from `http://en.wikipedia.org/wiki/Kernel_density_estimation`

Histograms and the classical approach

- Histogram:

$$\begin{aligned} \text{binning} : \mathbf{R} &\rightarrow \mathbf{Z} \\ x &\mapsto \text{int}\,(x/\delta x)\text{th bin} \end{aligned}$$

- Converting spike trains to words:

$$\begin{aligned} \text{discretization} : \text{space of spike trains} &\rightarrow \mathbf{Z} \\ r &\mapsto w \end{aligned}$$

What we need for kernel density estimation.

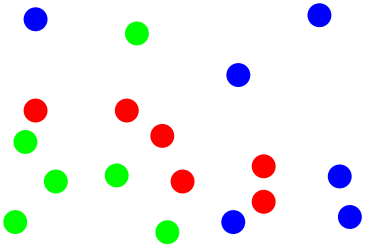
$$p(x) = \frac{1}{n}\sum_i k(x - x_i)$$

So we must have

$$\int k(x)dx = 1$$

That means we need to be able to integrate.

Spike train space.



Integrating in the space of spike trains I.

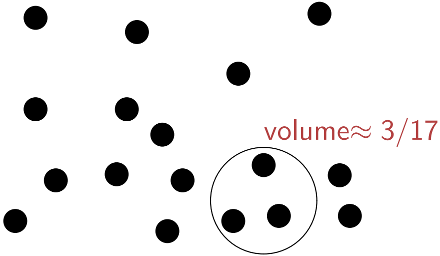
The space of spike trains has no coordinates, but it does have a measure given by the distribution of spike trains.

$$\mathrm{vol}(\mathcal{D}) = P(r \in \mathcal{D})$$

which can be estimated by the fraction of responses in $\mathcal{D}$

$$\mathrm{vol}(\mathcal{D}) \approx \frac{\#\text{ spike trains in } \mathcal{D}}{\#\text{ spike trains}}$$

Integrating in the space of spike trains II.



Basic idea

Want to calculate:

$$I(R;S) = \sum_{s\in\mathcal{S}} \int_{\mathcal{R}} p(r,s) \log_2 \frac{p(r|s)}{p(r)} dr$$

- Use $p(r)$to give a measure.
- Use KDE to estimate $p(r|s)$.
- Any integrals will be estimated by counting points.
- Kernels will be defined using volume.

Tobin and Houghton (Entropy 2013)

Result

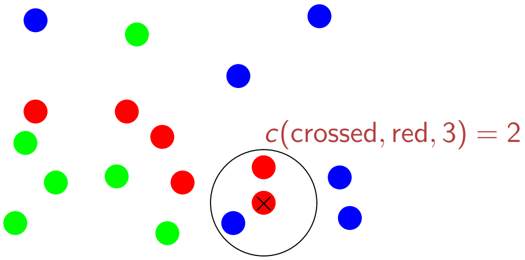
After some fiddling this give

$$I(R;S) \approx \frac{1}{n_r}\sum_i \log_2 \frac{n_s c(r_i, s_i; n_h)}{n_h}$$

- n_r number of responses.
- n_s number of stimuli.
- n_h size of the kernel.
- $c(r_i, s_i; n_h)$ number of responses to stimulus s_i in the kernel around r_i.

Tobin and Houghton (Entropy 2013)

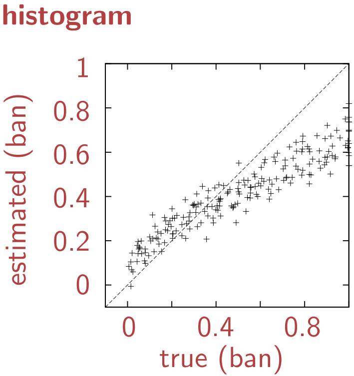
$c(r_i, s_i; n_h)$

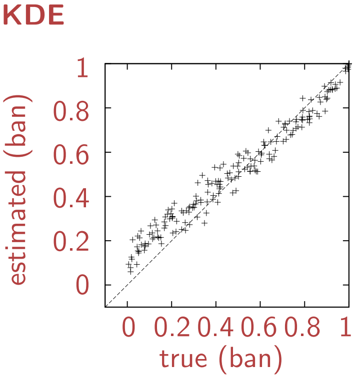


Results I

We are interested in the structure of spike train space as a metric space, so simulated test data was constructed using the sort of distribution of spike trains in metric space observed in Gillespie and Houghton (JCN 2011).

Results II

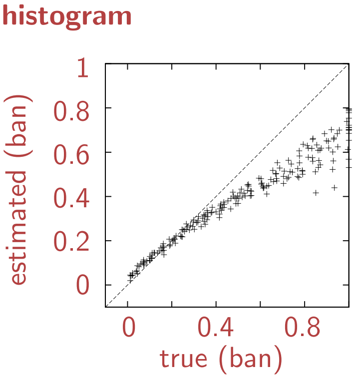


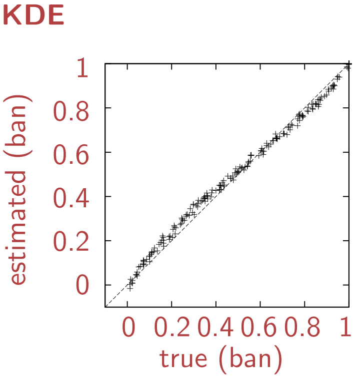


$n_s = 10$ and $n_t = 10$.

Tobin and Houghton (Entropy 2013)

Results III





$n_s = 10$ and $n_t = 200$.

Tobin and Houghton (Entropy 2013)

KDE and kNN

- KDE

$$I(R;S) \approx \frac{1}{n_r}\sum_i \log_2 \frac{n_s c(r_i, s_i; n_h)}{n_h}$$

- kNN

$$I_e(R;S) \approx F(n_k) + F(n_t n_s) - F(n_t) - \frac{1}{n_r}\sum_i F[C(r_i, s_i; n_k)]$$

  - They use a Kozachenko and Leonenko estimator and, by a clever choice of how they pick k for different parts of the estimate, they get all the volume-based terms to cancel.

Kraskov, Stöbauer and Grassberger (PRE 2004)

Euclidean metric

Picture from Google maps

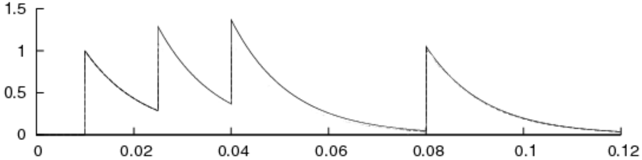
Non-Euclidean metric

Picture from Google maps

Non-metric

Picture from http://en.wikipedia.org/wiki/Uffizi

van Rossum metric



Spike trains mapped to functions and a metric on the space of functions induces a metric on the spike train space.

van Rossum (Neural Comp. 2001)

Multi-unit van Rossum metric

- There is a multi-unit easily computed version of the van Rossum metric.
- It relies on a time constant and a population parameter.

Houghton and Sen (Neural Comp. 2008) / Houghton and Kreuz (Network 2012)

The Victor-Purpura metric I.

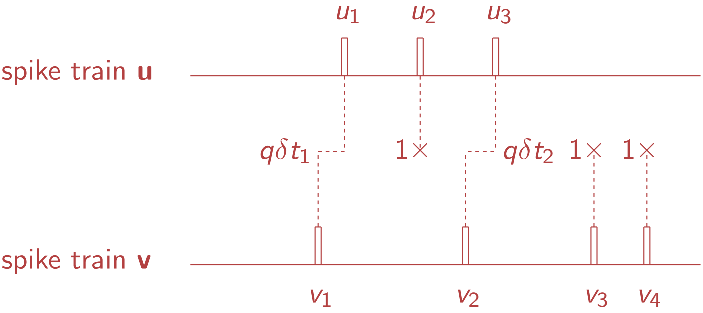
Edit one spike train in to the other:

1. Insertion of a spike with a cost of one.
2. Deletion of a spike with a cost of one.
3. Moving a spike a distance δt costs $q|\delta t|$.

The distance is the cost of the cheapest edit.

Victor and Purpura (J. Neurophys. 1996)

The Victor-Purpura metric II.



$$d = 3 + q(\delta t_1 + \delta t_2)$$

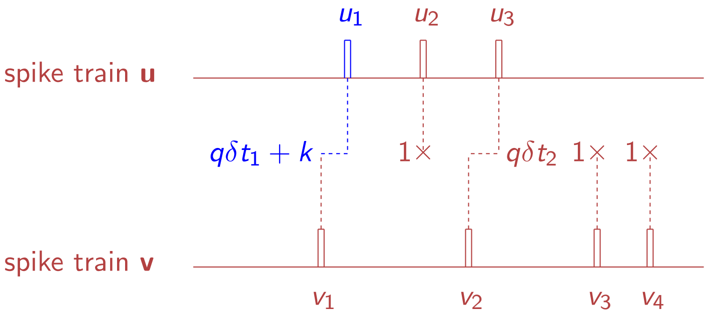
Multi-neuron Victor-Purpura I.

- The Victor-Purpura metric can be extended to measure a distance between a pair of population responses by adding a cost k for changing the identity of a spike.
- Edit one spike train in to the other:
  1. Insertion of a spike with a cost of one.
  2. Deletion of a spike with a cost of one.
  3. Change the neuron label of a spike at a cost k.
  4. Moving a spike a distance δt costs $q|\delta t|$.

  The distance is the cost of the cheapest edit.

Aranov et al. (J. Neurophys. 2004)

Multi-neuron Victor-Purpura I.



$$d = 3 + q(\delta t_1 + \delta t_2) + k$$

Labelled-line and population codes

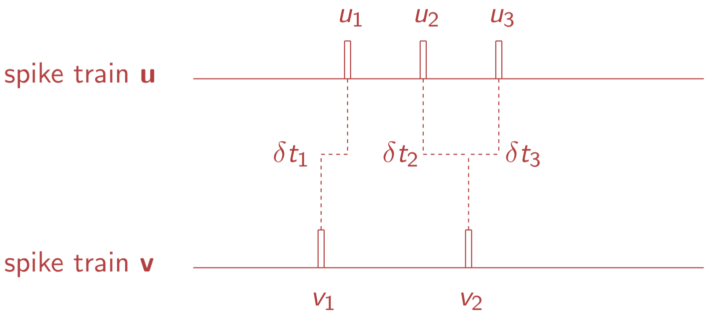
- If $k = 0$ it does not matter what neuron a spike came from, this is *a population code*.
  - It's like superimposing all the spike trains in the population and then working out the distance.
- If $k = 2$ it is never worth changing the label, this is *a labelled-line code*.
  - The distance between the two population responses is just the sum of the distances for the individual pairs of responses for each neuron.

Aranov et al. (J. Neurophys. 2004)

Problems

- Multi-unit VP metric computationally expensive.
- How do we calculate q and k?

SPIKE metric.

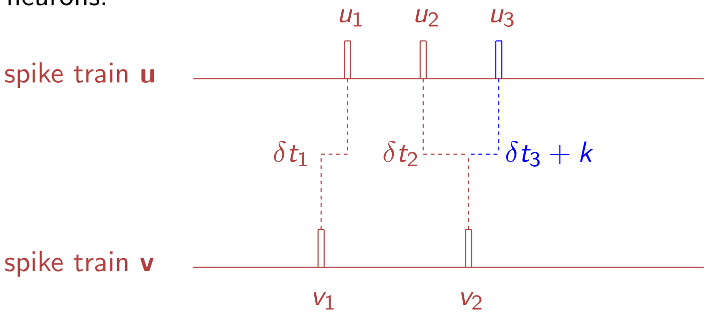


$$d = (\delta t_1 + \delta t_2 + \delta t_3) + (\delta t_1 + \delta t_3) = 2\delta t_1 + \delta t_2 + 2\delta t_3$$

Kreuz, Chicharro, Houghton, Andrzeja and Mormann (J. Neurophys. 2013)

Multi-neuron SPIKE metric.

Extending SPIKE to multiple neurons is easy, just add a ‘distance’ between neurons.



$$d == 2\delta t_1 + 2\delta t_2 + \delta t_3 + k$$

Multi-neuron SPIKE metric - problem.

Still stuck with k.

Conclusions

- Distance-based measures of mutual information.
- Multi-neuron distance functions.

Acknowledgements

- Collaborators.
    - Thomas Kreuz / Josh Tobin / James Gillespie / Cathal Cooney.
- Funding: the James S McDonnell Foundation.