Benchmark Data

Ciira wa Maina *et al.*

A key component of the convolved Gaussian process approach to model pol-II dynamics involves the estimation of delay between time series observed at different segments of the gene. To determine how well the convolved GP approach estimates delay, we investigated the performance of five delay estimation methods, namely cross-correlation (Corr), discrete correlation function (DCF) [1], the kernel approach of [2] (Kern), a GP approach with no convolution (GP-NoConv), and the convolved GP approach (GP-Conv).

In the convolved GP approach to model pol-II dynamics, we model the pol-II occupancy $y_i(t)$ in each gene segment $i \in \{1, \ldots, I\}$ as the convolution of a latent process $f(t)$ which is shared by all segments and a (possibly delayed) smoothing kernel $k_i(\tau - D_i)$ corrupted by an independent white Gaussian noise process $\epsilon_i(t)$ with zero mean and variance σ_i^2 [3, 4]. That is

$$y_i(t) = \alpha_i \int_{-\infty}^{\infty} f(t-\tau) k_i(\tau - D_i) d\tau + \epsilon_i(t), \tag{1}$$

where α_i is a scale factor and D_i is the delay of each segment. The latent process $f(t)$ is modeled as a random function drawn from a GP with zero mean and a squared exponential covariance function. The smoothing kernel is assumed to be Gaussian, that is

$$k_i(\tau) = \frac{1}{\sqrt{2\pi}\ell_i} \exp\left(-\frac{\tau^2}{2\ell_i^2} \right). \tag{2}$$

We used synthetic data to assess the performance of the convolved Gaussian process approach to delay estimation. To generate the synthetic data, the latent function $f(t)$ was given as a sum of Gaussian kernels. That is

$$f(t) = \sum_{i=1}^{N} \beta_i \exp\left(-\frac{(t-c_i)^2}{\sigma_i^2} \right).$$

N was fixed at 20 and the observation interval $t \in [0, 10]$. β_i, σ_i and c_i were generated at random with $\beta_i \in [0, 1]$, $\sigma_i \in (0.5, 1.5]$ and $c_i \in [2.5, 5]$. A random delay $D \in [1, 2.5]$ was used to generate the observations which were corrupted by additive Gaussian noise with $\sigma_n = 0.001$. To determine the effect of number of observations on the quality of inference we compute the median normalised square error (MNSE) of the estimated delay $\frac{\|D-\hat{D}\|_2^2}{\|D\|_2^2}$ as a function of the number of observations for 50 random realisations of the the signals. We also investigated the effect of distorting the shape of the observed signals by introducing convolution. In real signals the restriction that the shape remains unchanged sometimes leads to poor results. The parameters of the smoothing kernel in equation (1) were generated at random with $\alpha_i \in [0, 1]$ and $\ell_i \in (0.625, 2.5]$.

The synthetic observations are given by the convolution of $f(t)$ and the Gaussian kernel. This convolution can be evaluated analytically. We get

$$y_i(t) = \alpha_i \sum_{j=1}^{N} \beta_j \frac{\sigma_j}{\sqrt{2\ell_i^2 + \sigma_j^2}} \exp\left\{ -\frac{1}{2} \frac{(t - c_j - D_i)^2}{\ell_i^2 + \sigma_j^2/2} \right\} + \epsilon_i(t)$$

Results

Code to generate the synthetic data and test the five methods is freely available from `https://github.com/ciiram/PyPol_II`. The main program is Toy_data.py. To run the program, start an IPython shell and type run Toy_data.py [conv] [num_obs] [num_trials] or type python Toy_data.py [conv] [num_obs] [num_trials] at the command line. The numpy, scipy and pylab libraries need to be installed. The command line arguments are conv which is 1 if the synthetic data is to be distorted via convolution and zero otherwise. The number of observations num_obs and the number of random data realisations num_trials.

Running run Toy_data.py 0 6 50 produces the output

```
#######################
# Experiments Running #
#######################
Num Obs  6  Trial:  1
...
Num Obs  6  Trial:  50
####################
#      Results     #
####################
Method     MNSE
Corr       0.035456
DCF        0.029834
Kern       0.003968
GP-NoConv  0.001622
GP-Conv    0.002172
```

We see that without convolution the kernel approach works well but is outperformed by the GP based techniques.

Running run Toy_data.py 1 6 50 introduces convolution and produces the output

```
#######################
# Experiments Running #
#######################
Num Obs  6  Trial:  1
...
Num Obs  6  Trial:  50
####################
#      Results     #
####################
Method     MNSE
Corr       0.031643
DCF        0.036698
Kern       17.104576
GP-NoConv  0.000156
GP-Conv    0.000056
```

We see that in this case the kernel method performs poorly. The GP based approaches still perform the best with GP-Conv outperforming GP-NoConv as expected.

References

[1] Edelson RA, Krolik JH (1988) The discrete correlation function - A new method for analyzing unevenly sampled variability data. The Astrophysical Journal 333: 646-659.

[2] Cuevas-Tello JC, Tino P, Raychaudhury S (2006) How accurate are the time delay estimates in gravitational lensing? Astronomy and Astrophysics 454: 695-706.

[3] Boyle P, Frean M (2005) Dependent Gaussian processes. In: In Advances in Neural Information Processing Systems 17. MIT Press, pp. 217–224.

[4] Alvarez M, Lawrence ND (2008) Sparse Convolved Gaussian Processes for Multi-output Regression. In: NIPS. pp. 57–64. URL `http://books.nips.cc/papers/files/nips21/NIPS2008\_0170.pdf`.