# V1 data: larger nBasis for X

Window = 10 for all v1 fitting

nBasis for X = 20: check if reasonable



Under nBasis for X = 20, fit different models

|  |  |  |  |
| --- | --- | --- | --- |
|  | Adaptive CMP  nBasis\_X = 20,  nBasis\_G = 20 | Adaptive CMP  nBasis\_X = 20,  nBasis\_G = 10 | Adaptive CMP  nBasis\_X = 20,  nBasis\_G = 5 |
| Q | Qopt1 =  1.0e-04 \*  0.2438 0.1533 0.1935 0.3604 | Qopt1 =  1.0e-03 \*  0.0340 0.0095 0.0371 0.1866 | Qopt =  1.0e-04 \*  0.3408 0.0704 0.2648 0.4711 |
| Parameter trace |  |  |  |
| Llhd/trial: training (ratio to static Poisson) | -83.7555 (5.6387) | -83.3615 (6.0327) | -84.4151 (4.9791) |
| Llhd/trial: held-out (ratio to static Poisson) | -88.1554 (2.4252) | -88.5377 (2.0429) | -87.8594 (2.7212) |
| fitting |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  | Adaptive CMP  nBasis\_X = 20,  nBasis\_G = 1 | Adaptive CMP  nBasis\_X = 20,  constant nu | Adaptive Poisson  nBasis\_X = 20 |
| Q | Qopt=  1.0e-04 \*  0.3430 0.1059 0.4189 | Qopt3 =  1.0e-04 \*  0.2840 0.1400 | Qopt4 =  1.0e-04 \*  0.2781 0.1439 |
| Parameter trace |  |  |  |
| Llhd/trial: training (ratio to static Poisson) | -84.6182 (4.776) | -84.8997 (4.4945) | -87.2502 (2.144) |
| Llhd/trial: held-out (ratio to static Poisson) | -87.7935 (2.7871) | -87.8069 (2.7737) | -89.4377 (1.1429) |
|  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | CMP  nBasis\_X = 20,  nBasis\_G = 20 | CMP  nBasis\_X = 20,  nBasis\_G = 10 | CMP  nBasis\_X = 20,  nBasis\_G = 5 | Poisson  nBasis\_X = 30 |
| Llhd/trial: training (ratio to static Poisson) | -87.2839 (2.1103) | -87.4785 (1.19157) | -87.5004 (1.8938) | -89.3942 |
| Llhd/trial: held-out (ratio to static Poisson) | -88.2715 (2.3091) | -88.4001 (2.1805) | -88.3753 (2.2053) | -90.5806 |

nBasis for X = 30:



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Adaptive CMP  nBasis\_X = 30,  nBasis\_G = 30 | Adaptive CMP  nBasis\_X = 30,  nBasis\_G = 20 | Adaptive CMP  nBasis\_X = 30,  nBasis\_G = 10 | Adaptive CMP  nBasis\_X = 30,  nBasis\_G = 5 |
| Q | Qopt =  1.0e-04 \*  0.2372 0.1093 0.0905 0.7797 | Qopt =  1.0e-04 \*  0.2351 0.1048 0.0704 0.7653 | Qopt =  1.0e-04 \*  0.2954 0.0340 0.2200 0.7038 | Qopt =  1.0e-04 \*  0.3141 0.0519 0.1844 0.3860 |
| Parameter trace |  |  |  |  |
| Llhd/trial: training (ratio to static Poisson) | -82.7905 (6.3221) | -83.201 (5.9166) | -83.9318 (5.1808) | -84.2835 (4.8291) |
| Llhd/trial: held-out (ratio to static Poisson) | -88.1956 (2.0459) | -87.9976 (2.2439) | -87.7604 (2.4811) | -87.5369 (2.7046) |
| fitting |  |  |  |  |

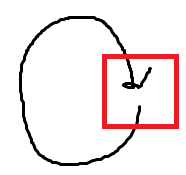
|  |  |  |  |
| --- | --- | --- | --- |
|  | Adaptive CMP  nBasis\_X = 30,  nBasis\_G = 1 | Adaptive CMP  nBasis\_X = 30,  constant nu | Adaptive Poisson  nBasis\_X = 30 |
| Q | Qopt2 =  1.0e-04 \*  0.3216 0.0939 0.3310 | Qopt3 =  1.0e-04 \*  0.2877 0.1270 | Qopt4 =  1.0e-04 \*  0.2972 0.1314 |
| Parameter trace |  |  |  |
| Llhd/trial: training (ratio to static Poisson) | -84.3786 (4.734) | -84.5723 (4.5403) | -86.9611 (2.1515) |
| Llhd/trial: held-out (ratio to static Poisson) | -87.52 (2.7215) | -87.5636 (2.6779) | -89.1008 (1.1407) |
| fitting |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | CMP  nBasis\_X = 30,  nBasis\_G = 30 | CMP  nBasis\_X = 30,  nBasis\_G = 20 | CMP  nBasis\_X = 30,  nBasis\_G = 10 | CMP  nBasis\_X = 30,  nBasis\_G = 5 | Poisson  nBasis\_X = 30 |
| Llhd/trial: training (ratio to static Poisson) | -86.9819 (2.1307) | -87.1264 (1.9862) | -87.2809 (1.8317) | -87.3048 (1.8078) | -89.1126 |
| Llhd/trial: held-out (ratio to static Poisson) | -87.9594 (2.2821) | -88.0711 (2.1704) | -88.1756 (2.0659) | -88.1597 (2.0818) | -90.2415 |

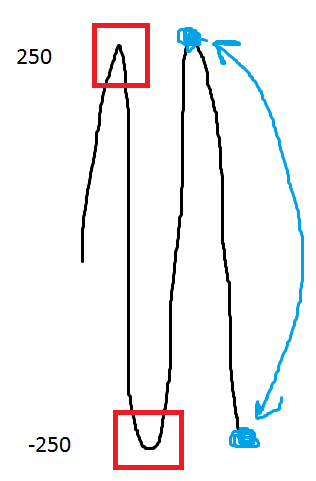
Well... It seems that nBasis\_G = 1 is the best for both nBasis\_X = 20 and 30. Or show the MSE (in terms of CMP\_mean and observed counts)?

# Hippocampus data

I’m confused about the circular basis… For V1, using circular basis makes sense for me. Since it’s about circulation in angle, and the state vectors within the red rectangle below should be similar.



However, for hippocampus data, it’s about circulation in a linear track position. The state vectors within red rectangles surely should be similar, but the circular basis tries to stick 2 end positions (blue dots) together. In other words, the experiment is a linear track, but I think the circular basis force the track to be a circle.



In practice, when using the circular basis, GLM, CMP regression and adaptive filtering will always fail (ill-conditioned X/ non-finite objective function).

The following results make sense only if you agree with my argument for non-circular basis.

Neuron = 10, bin = 10. Code: demo3\_tmp.m in demo/hc. Window = 50 for all.

|  |  |
| --- | --- |
| nBasis for lambda = 1, nBasisi for nu = 1  Qopt =  1.0e-03 \*  0.0056 0.5000  llhd =  -3.5941e+03 | nBasis for lambda = 4, nBasis for nu = 1  Qopt =  1.0e-03 \*  0.0045 0.1122 1.0000  llhd =  -3.1957e+03 |
|  |  |
| nBasis for lambda = 20, nBasis for nu = 1  Qopt =  1.0e-03 \*  0.1400 0.0981 0.9970  llhd =  -3.0361e+03 | nBasis for lambda = 4, nBasis for nu = 4  Qopt =  1.0e-03 \*  0.0034 0.0507 0.9997 0.9993  llhd =  -3.1604e+03 |
|  |  |
| nBasis for lambda = 20, nBasis for nu = 4  Qopt =  1.0e-03 \*  0.0986 0.0878 0.1000 0.1000  llhd =  -3.1230e+03 | nBasis for lambda = 30, nBasis for nu = 1  Qopt =  1.0e-03 \*  0.0159 0.1922 0.7304  llhd =  -2.9960e+03 |
|  |  |

When we don’t bin the data, the fitting is not good.

# Three cases revisit

All the Q selection is done when the “window” is turned off (i.e. window size = 1). So, the Q tuned results are the same for different window size. The reason is that when the “window” is turned on, the Q estimation will be biased (overestimate for “forward” window and underestimate for “backward” & “centering” window). This is shown by precious results in “explain1.docx”.

To investigate the performance of Q-tuning, I use the example in “demo\_QTune.m”. Now I set the forward and backward window size be 10, and centering window size be 11 (5 forward + 5 backward). The code can be found in “demo\_QTune\_window.m” In the following plots, the 1st column show heatmaps, the second column show fitting under true Q and tuned Q. The true Q is diag([1e-4 1e-5]).

|  |  |  |
| --- | --- | --- |
|  | Heatmap: prediction llhd | fitting |
| Forward |  |  |
| Backward |  |  |
| Centering |  | Similar to “backward” one. Didn’t run. |

For these 3 cases, saying “these might just be too little data to really estimate Q or the window size” make sense to me. So for these 3 cases, I now just use some pre-set Q (1e-3 for all 3 cases) and window size (50 for case 1 & 2, 100 for case 3).

|  |  |  |
| --- | --- | --- |
| Case 1 | Case 2 | Case 3 |
|  |  |  |

The ylim are set as [round(min(true\_value)) – 5 round(max(true\_value)) + 5] for all plots.