

# Playing around with parameters

## Optimizer

### STLSQ

*"Sequentially thresholded least squares"*

They take the argument  $\lambda$ , which specifies a threshold "of sparsity". Should parameter  $p_i$  be smaller than  $\lambda$ , it will be removed (and its corresponding term)

Meaning that the bigger the threshold is, the **less** terms may be present in the final model. In other words, the bigger the threshold, the more terms will be removed.

For `1D-Neuron-Multiple` the value of  $\lambda$  seemed to be best around  $(10^{-1}, 10^{-3})$ , smaller values of  $\lambda$  included insignificant noise.

“ Default value is  $\lambda = 10^{-1}$

Also for the [basic example](#) they set the value to  $\lambda = 10^{-1}$ , which seems to agree.

TODO: STLSQ with vector of thresholds

## Metrics

### $L_2$ Norm error

$L_2$  (or  $L_2$ ) norm of the error gives the  $L_2$  of an error in each dimension

Of course the smaller the better

# TVDIFF

Most times around 300 iterations seems to be by far enough

## Regularization $\alpha$

The regularization parameter  $\alpha$  tells us how strongly should the derivative be regularized (think of it as smoothed)

The bigger the  $\alpha$  the less it oscillates, though the less "features" of the true derivative it really exhibits

It is mostly visible when there is a big spike in the derivative. Then the `tvdiff` is unable "to catch up" when strongly regularized doesn't handle the spike well (it simply doesn't feature nearly as big of a spike)

I'd recommend starting with a higher  $\alpha$  and slowly increasing it, until we find the derivative smooth enough.

With small  $\alpha$  always check if it more or less corresponds to data (it has tendency to oscillate when the function is too constant)

## Epsilon $\epsilon$

Using `tvdiff` with  $\epsilon=1e-9$ , we obtain a strongly regularized result. Larger values of  $\epsilon$  improve conditioning and speed, while smaller values give more accurate results with sharper jumps.

## Scale and preconditioner

TO BE DONE

# Performance

In general the more data (and thus the derivation) varies in scale, the worse to model performs

# Collocations

Data collocation is only used when derivative is NOT supplied (and is surely better than forward diff)

TODO: Usage collocations on existing derivative?

---

Revision #1

Created 28 December 2022 14:28:32 by Sceptri

Updated 28 December 2022 15:29:38 by Sceptri