SINDy: Notes and Thoughts

September 2018

* Sparsity
  + SINDy relies on the assumption that the underlying system governing equations *f* are sparse in some parameter space.
  + If *f* isn't sparse, but sparse representation is good enough, sparsity knob lambda will threshold the allowed number of terms? Or coefficient values?
  + What is the intuition for choosing a range of lambda to test?
  + Go back and look at the math of exactly how lambda works.
  + SINDy is not robust to lambda space resolution.
    - What resolution is necessary to correctly identify the model?
    - Test model identification convergence with increasing lambda resolution.
* Coefficient values
  + What sets the threshold for these (in the sparsification routine)?
  + The thresholding method would seem to require that the range of the coefficient values *and* the effects on their corresponding feature in the governing equations are commensurate across all features. How is this dealt with?
    - Standardization? How do you know to do that ahead of time and how to do it?
    - There is normalization of theta at some point… go back and look at it again.
  + Can we calculate a metric of feature importance and feature sensitivity? Because the output model is deterministically specified, this should be trivial, no?
  + In Niall's implementation, specified coefficient values outside a certain range break SINDy's ability to correctly identify the model.
    - Why is this?
    - How can we make SINDy robust to this?
* Parameter space
  + How does one find optimal transformations of variables to meet the sparsity assumption? i.e. How do you best transform parameter space for use in SINDy?
  + Go back and look more carefully at the theoretical underpinnings of time delay embedding and accompanying parameter space transformation/de-transformation.
* Derivatives
  + In the case of discrete observations (which is everything in the real, not in a lab, world), SINDy relies on a "sufficiently accurate" method for calculating the derivative given the level of noise in the data. How do you evaluate sufficiently accurate?
  + Why does Niall specify the derivative the way she does and why does it work?
* Rank
  + Why does redundant (overdetermined?) information (equation) break SINDy?
  + How do I make a workflow that is robust to that?
  + Independence testing prerequisite to SINDy?
* SINDy not robust to:
  + Lambda space resolution.
  + Specified coefficient values.
  + Dependent specified equations.
* Possible improvements
  + Address the "not robust" failures with method changes and/or built in tests/warnings/error messages.
  + Look at models near the lowest AIC to get direction for further refinements.

Timescale

* Task:
  + Fast exogenous (e.g. precipitation) observations with slow endogenous (e.g. health metric) observations.
  + How to work across these timescales?
* Options:
  + Bespoke aggregation:
    - Determine your own aggregation of the fast variable to the slow variable timescale.
    - Include that aggregated time series as an exogenous variable in a SINDy at the slow timescale.
  + Discovering the (simple additive) memory kernel:
    - Includepast (fast timescale) time steps as features in a SINDy at the slow timescale.
    - The number of timesteps one can include as a feature independent of the number of fast timesteps per slow timestep.
    - Number of past timesteps included potentially limited by tractability.
    - The difference in timescales is now irrelevant for the SINDy.
    - Although, if you are just adding single timestep non-interacting fast timestep features, you are only modifying the "1" term of the specified model.
  + Layered SINDy:
    - SINDy with LHS = feature from slow timescale SINDy, and RHS = past timesteps of fast variable.
    - The specified fast timescale SINDy result then gets passed to the slow timescale SINDy.
    - What is the most efficient way to do this?
* Considerations:
  + How do the timestep approaches described above interact with time delay embedding? Review math and SINDy implementation.
  + Looking at timescale/observation intervals links into system memory theory/analysis.

Theoretical understanding gaps (review lit, SI, and implementation packages):

* Why redundant/overdetermined information breaks SINDy.
* Differences in derivative calculation implementations in Mangan vs. Brunton.
* Current theory/implementation of time delay embeddings and necessary parameter space transformations.
* The role of normalization in the SINDy routine.
* How lambda and coefficient thresholding work in the SINDy routine.

Software notes:

* Found and corrected an error in the Mangan et al. 2017 software package: Information criteria vs. Number of terms plot was plotting relative AIC instead of the corrected relative AICc.