Lt David Crow

Activity Report 1

24 March 2019 – 6 April 2019

Week 1

* Skimmed relevant sections of Captain Stone’s dissertation
* Spoke briefly with Dr. Borghetti about problem domain
* Installed C++ EDM, Python EDM, and R EDM
  + I would prefer C++, but it’s lacking in documentation
  + Python seems more promising, but it too is a bit lacking in documentation
  + R is well-documented, but I’ve never worked with the language or with CRAN
* Gathered sources (9)

Week 2

* Update: definitely going to use pyEDM — easier to work with and meshes well with CSCE 723/823
* Met with Captain Stone
* Reference *Research Notes* for more details
* Met with Dr. Borghetti
* Reference *Research Notes* for more details
* Gathered sources (6)

Research Notes

* 2 April 2019 meeting with Captain Stone
  + Semi-supervised dataset labeling
    - Create database of directed, weighted graphs
      * Weighted by causality, for example
    - Train on database
    - Test on partially-labeled subsection of database
    - Reference pg. 114 of his dissertation
      * Given Speed and RPM clusters, can we label the unknown cluster with brake?
      * Causalities hand-calculated with rEDM on one trial of Stone’s data
    - If toy problem is successful, apply model to car data
    - Need to speak to Dr. Borghetti about assigning weights (e.g. strongest weight, weakest weight, or mean weight) to the links between clusters
  + Reverse engineering pipeline
    - <https://github.com/brent-stone/CAN_Reverse_Engineering>
    - Bugs in /Pipeline, so use /Pipeline\_multi-file
      * Can I fix the bugs in /Pipeline? Debug hex → Pandas → int → plot code
    - Given text file of hex data, pipeline should generate scatterplots/time-series (including some clusters labeled by J1979)
* 2 April 2019 meeting with Dr. Borghetti
  + Need to identify multiple features (e.g. 0.5s intervals, standard deviation, mean) to effectively train a model
  + Directed, weighted graph not all that necessary
  + Learning on a bunch of time-series plots is not normal
* 2 April 2019 lingering thoughts
  + Process to answer semi-supervised dataset labeling
    - Collect data and use Captain Stone’s pipeline to extract plots
    - Label known plots using J1979
    - Apply EDM/clustering techniques to identify various clusters
      * These are nodes in the graph
    - Link the nodes by calculating edge weights
      * What’s the best way to do this?
    - Train on labeled instances of the graph
      * Ideally, model will learn based on labeled nodes and causality between various nodes
    - Test partially-labeled graph
    - Probably should apply the above process to a fully-labeled toy problem before attempting car data problem
  + Other possible projects
    - Route reconstruction with limited data
      * Utilize tire pressure IDs
      * Limited number of sensors (e.g. key intersections, neighborhood entrances)
      * Need to research
        + Update: it’s already done
    - Route prediction based on previously-collected data
      * Can we learn this?
        + Classifier (e.g. home, gas station, grocery store, work)
      * Consider predicting other factors, like collisions and bad traffic
      * Need to research
        + Update: it’s already done
* 4 April 2019 meeting with Dr. Borghetti
  + Asked whether the semi-supervised dataset labeling problem is more applicable to deep learning
  + After discussing further, he decided that SSDL is feasible under machine learning
    - Difficulties
      * Obtaining the fully-labeled graphs (however, we can generate the dataset ourselves)
      * Representing each graph (each observation) as a feature — how do we do this?
      * Outputting the y-value — what if multiple nodes are unlabeled?
      * In other words, knowledge representation
    - Ideas
      * Use matrices to represent the graphs (RPM/Speed/etc. by RPM/Speed/etc., where cell values are the causality weights)
    - Going forward
      * Consider other graph-labeling techniques
        + Granger causality
        + EDM
      * Problem only concerns a little bit of machine learning
        + For the thesis, this is fine