Weekly Progress: 14 July - 20 July 2014

July 21, 2014

All of the source code for this work can be found on GitHub at:

http://github.com/cbuntain/ChangePointDetection

1 Achievements

- **Extended our data simulator to produce higher-order VAR models** – To test our algorithms' sensitivity to deviations in the underlying model, we needed methods to generate more than just first-order VAR models. To accomplish this task, however, we needed to extend our ability to generate Φ matrices that fulfilled the necessary conditions (i.e., that the roots of the determinant of the VAR polynomial lied outside the unit disk).

  - **Derived mathematical relationships between higher-order Φ matrices and lag polynomial roots** – In the previous week, we derived the relationship between the roots of a VAR(1) model and the properties of the candidate matrix: roots of $|I - \lambda\Phi|$ are equal to λ^{-1}, where λ is an eigenvalue of the matrix. For higher-order polynomials, we can leverage this derivation to find the roots of $|(I - \lambda A_1)(I - \lambda A_2)...(I - \lambda A_p)|$. Then, the Φ matrices become combinations of the A matrices.

  - **Implemented the $A \rightarrow \Phi$ expansion for second- and third-order VAR models** – Expanded the A polynomial to calculate second- and third-order Φ matrices, and integrated those models into the data simulator.

- **Tested LRT, CUSUM, and KCD algorithms against higher-order VAR models** – Using the test harness developed the previous week, we ran experiments over three VAR model orders, varying dimensionality and change-point count in each. As expected, as VAR order increases, detection accuracy decreases slightly in all three algorithms, and false positives increase significantly.

- **Compared LRT, CUSUM, and KCD's ability to detect covariance shifts and mean shifts** – We extended the test harness to support generating covariance change points or mean shift change points and compared performance between the three algorithms. LRT seems to perform as well as or better than CUSUM and KCD in both classes of change point in VAR(1), VAR(2), and VAR(3).

- **Derived loose bounds on the approximate scale of change points KCD could detect** – The KCD algorithm has four parameters: window size m, data variance γ, the proportion of data expected to be outliers ν, and a threshold for the KCD statistic η. If r_f is the first index in the time series of where the change point starts, and r_l is the last index when the change point is complete, for KCD to be able to detect this change point, the range in which the change point occurs $r_l - r_f$ likely should be less than $2m(1 - \nu)$. Otherwise, the ability to differentiate regimes would decrease as the change point range increased.

- **Ran LRT, CUSUM, and KCD against the Bitcoin and bridge data** – We successfully processed both data sets using all three algorithms, though we have not yet compared the data to find overlapping change points or change points correspond to real events.

- **Implemented a prototype version of a density-based change point detection algorithm** – We implemented a prototype version of a change point detection algorithm that uses density ratios to

delineate between regimes. The implementation follows Kawahara and Sugiyama's paper, "Sequential Change-Point Detection Based on Direct Density-Ratio Estimation."

- **Initiated a test of LRT, CUSUM, and KCD against provided data sets** – Using the Matlab/R-generated data sets, we began a series of experiments comparing LRT, CUSUM, and KCD (using a sliding window for covariance in KCD rather than the actual data). This experiment is currently still running.

2 Plans for the Upcoming Week

- **Refine density ratio-based implementation** – We need to finish this implementation and integrate it into our existing test harness.

- **Align Bitcoin and bridge data change points** – Given the change points we've detected, we need to determine overlap between the algorithms and potential exogenous reasons for the change points identified.

- **Summarize and present our work to date** – To support knowledge transfer, we will compile a summary and presentation of our on change point detection to present to the SNE group on Thursday.