Tutorial for STADIA Classification Stage using the "Diagnostic Mode"

Tutorial last updated on October 22, 2021

The diagnostic mode built into STADIA is an opportunity for the user to contribute experience and domain knowledge into the classification process as the segmented DI length history data is separated into cluster groups. This document serves as a tutorial for operating STADIA in the diagnostic mode, and how to interpret the output in selecting the number of clusters to use in the classification stage to run the full, automated mode of STADIA. Here, the gap statistic is used to help identify the optimal number of clusters, the *k*-value, that should be used to separate the data considered in each scenario (Tibshirani, Walther, and Hastie 2001)**.** Note that for this tutorial, the user is expected to be familiar with the STADIA stages, and to understand that the segmentation stage is completed at the point where the diagnostic mode analysis takes over, such that the clustering is performed on points in 3-dimensional space each representing the three line segment characteristics identified from the length history data.

Gap statistic analysis

In all of the variations of the diagnostic mode, the *k*-means algorithm is repeated 100 times for each of the *k*-values in the range 1 to 12. This range of *k*-values covers beyond the expected range useful in practice, but is included for completeness and to aid users in determining the best *k*-value that makes sense for them, as is the intention of the diagnostic mode. For each run, the gap statistic is measured, and the results are displayed as the gap statistic plot, indicating the mean gap value and standard error bars for each *k*-value. The gap statistic is a comparison of the within-cluster dispersions measured for the data in question and a "null" reference data set drawn from a random uniform distribution. Larger gap statistic values imply better clustering results. In their paper, Tibshirani et al. describe the gap statistic criterion, which suggests that the optimal number of clusters is the smallest *k*-value such that the following inequality is satisfied:

$$Gap(k) \geq Gap(k+1) - s_{k+1}$$

where $Gap(k+1)$ and s_{k+1} are the mean and the standard error of the gap statistic, respectively, when using (*k*+1)-many clusters (Tibshirani, Walther, and Hastie 2001)**.** Choosing the smallest *k*-value that satisfies this condition promotes choosing a lower number of clusters, which also prevents overfitting. Also, stricter interpretations can require choosing the first local maximum, which stems from the fact that choosing the optimal number of clusters is a difficult task considering an unsupervised machine learning approach, where the ground truth is unknown. For this reason, the user must use discretion and apply scientific domain expertise when choosing the final *k*-value.

Running the three diagnostic mode versions in STADIA

The diagnostic mode in STADIA can run the gap statistic analysis for *k*-means clustering in three ways: for **all segments**, for **positive slope segments** only, and for **negative slope segments** only. The corresponding user-defined parameters for running STADIA's classification stage using the different options in diagnostic mode are as follows:

- KMEANS_DIAGNOSTIC_ALLFLAG: an indicator for conducting gap statistic analysis using all of the segment points. Acceptable values are 1 (on) or 0 (off).

- KMEANS_DIAGNOSTIC_PosSlopeFLAG: an indicator for conducting gap statistic analysis using only the points corresponding to positive slope segments. Acceptable values are 1 (on) or 0 (off).
- KMEANS_DIAGNOSTIC_NegSlopeFLAG: an indicator for conducting gap statistic analysis using only the points corresponding to negative slope segments. Acceptable values are 1 (on) or 0 (off).

For each variation of the diagnostic mode, after conducting the gap statistic analysis and saving the corresponding output files, STADIA halts the classification stage by stopping the current run and displaying a warning message indicating the version of diagnostic mode that is turned on. This way only one diagnostic mode variant is run at a time.

In addition to the output files, STADIA also displays some information to the MATLAB terminal screen. Included is an automatically suggested *k*-value which appears in the dialogue printed to the terminal for each of the diagnostic mode options. This *k*-value is also indicated in title of the cluster plot, where the data are clustered using the suggested *k*-value. This automatically determined *k*-value corresponds to the most general gap statistic criterion. However, the user should use prior knowledge of their data and its characteristics when making a decision in the event that this automated suggestion is contradictory to the specific case scenario being explored. For example, if the user feels that the suggested *k*-value suggests too few clusters, the next local maximum may be the user's choice.

After a diagnostic run is completed, the user is expected to study the gap statistic results, make decisions and select STADIA variables accordingly, and move to the next diagnostic mode option. Once all the diagnostic mode options have been exhausted, the user should have all the information needed from the diagnostic mode, specifically the two *k*-values for the positive and negative slope segments. The user can then turn off the diagnostic modes by setting all flag variables listed above to 0, and execute the code again to run STADIA in the fully automated mode.

Diagnostics Mode Part 1: All Segments

The purpose of attempting to cluster **all segments** together is primarily for demonstration purposes, and is included as a STADIA option for completeness. When setting KMEANS_DIAGNOSTIC_ALLFLAG=1, or turning this mode on, gap statistic analysis is performed on the data comprised of all the segment points. STADIA developers regularly observed this approach to yield a gap statistic plot that is monotonically increasing, where a clear local maximum or a point satisfying the gap criterion is not evident or appears at a *k*-value larger than what is reasonably expected. This is an indicator that the data may have substructures not yet considered, which would benefit from considering subsets of the data separately. In other circumstances, if the identified *k*-value is optimal for the total number of clusters, the resulting clustering boundaries may not be the as good as performing the clustering on a subsets of the data separately. Thus, this variation in the diagnostic mode is merely for the user to verify that this indeed is not the best way to conduct the clustering procedure, and to justify the need to break up the data into smaller subsets prior to clustering.

Furthermore, there is no *k*-value associated to conduct clustering for all the segments together in automated mode. For this reason, the STADIA classification stage and the remaining part of the diagnostic

mode moves forward by clustering positive slope segments and negative slope segments separately. At this time, the user should switch off the "all segments" variation of the diagnostic mode by setting KMEANS_DIAGNOSTIC_ALLFLAG=0, and re-running STADIA to conduct the next step in the process, which is to analyze the positive slope segments.

Diagnostics Mode Part 2: Positive Slope Segments

Setting KMEANS_DIAGNOSTIC_PosSlopeFLAG=1 turns on the diagnostic mode for conducting the gap statistic analysis on the positive slope segment data. Typical results yield the optimal *k*-value to be between 1 through 3. Values greater than or equal to 4 are typically not interpretable and difficult to explain, though they are included as part of the gap statistic analysis in STADIA for completeness. At this time, the user should decide on the *k*-value between 1 through 3 that represents the number of clusters that best separates the positive slope segments. The user should assign that value to the KMEANS_NumClust_PosSlope variable in the "Input_and_Run.m" file. After completing this step, the user should switch off the "positive slope segments" variation of the diagnostic mode by setting KMEANS_DIAGNOSTIC_PosSlopeFLAG =0, and re-running STADIA to conduct the next step in the process, which is to analyze the negative slope segments.

Diagnostics Mode Part 3: Negative Slope Segments

Setting KMEANS_DIAGNOSTIC_NegSlopeFLAG=1 turns on the diagnostic mode for conducting the gap statistic analysis on the negative slope segment data. Typical results yield the optimal *k*-value to be between 1 through 3. Values greater than or equal to 4 are typically not interpretable and difficult to explain, though they are included as part of the gap statistic analysis in STADIA for completeness. At this time, the user should decide on the *k*-value between 1 through 3 that represents the number of clusters that best separates the negative slope segments. The user should assign that value to the KMEANS_NumClust_NegSlope variable in the "Input_and_Run.m" file. After completing this step, the user should switch off the "negative slope segments" variation of the diagnostic mode by setting KMEANS_DIAGNOSTIC_NegSlopeFLAG =0, and re-running STADIA to conduct the next step in the process, which is to run STADIA in the fully automated mode.

Special Considerations:

As mentioned in the positive and negative slope segment analysis sections, expected selections for the optimal *k*-values are the integers 1, 2, or 3. Several data sets tested so far have not yielded an optimal *k*-value greater than 3 (Mahserejian et al., n.d.). For this reason, the automated mode only accepts *k*-values no more than 3 for the KMEANS_NumClust_PosSlope and KMEANS_NumClust_NegSlope variables. Choosing anything larger than 3 will terminate STADIA early and return an error explaining the circumstance.

Another special circumstance exists when selected a *k*-value of 2 for either the positive or negative slope segment data subsets. Choosing *k*=1 is obvious, since all of the segment points would be categorized as growth or shortening, and the stutter label would be ignored. Choosing *k*=3 would assign the following labels to each cluster: up stutter, brief growth, and sustained growth for the positive slope segments; and down stutter, brief shortening, and sustained shortening for the negative slope segments. However, to

assign two of the possible three labels when a user selects *k*=2, the user needs to provide one of the following additional options, which determines how the relevant data subsets will be labeled.

So, if the user assigns KMEANS_NumClust_PosSlope = 2, then the STADIA variable KMEANS_Pos2_Option needs to be assigned one of the following values:

- KMEANS_Pos2_Option = 'A' : two growth phases, such that the two clusters will be assigned brief growth or sustained growth labels
- KMEANS_Pos2_Option = 'B' : one growth and one stutter phase, such that the two clusters will be assigned sustained growth or up stutter labels

If the user assigns KMEANS_NumClust_NegSlope = 2, then the STADIA variable KMEANS_Neg2_Option needs to be assigned one of the following values:

- KMEANS_Neg2_Option = 'A' : two shortening phases, such that the two clusters will be assigned either brief shortening or sustained shortening labels
- KMEANS_Neg2_Option = 'B' : one shortening and one stutter phase, such that the two clusters will be assigned either sustained shortening or down stutter labels

Output content:

For any variation of the diagnostic mode, results are saved into a dedicated directory for running diagnostic mode on a single input length history file. Within this directory

The diagnostic mode directory has the following attributes:

- Directory name criteria:
  - "STADIA_DiagnosticMode_Files_for_" as a prefix
  - Filename of the input length history file (without the file extension)
  - Example diagnostic mode directory name:
    - "STADIA_DiagnosticMode_Files_for_length_13PF_10uM_10hr"
    - Note: this example uses "length_13PF_10uM_10hr.txt" as the corresponding input file
- Subdirectory names for each type of diagnostic mode:
  - AllSegments_Diagnostics
  - PositiveSlopeSegments_Diagnostics
  - NegativeSlopeSegments_Diagnostics

For each diagnostic mode variant, the following output files are produced in the corresponding subdirectory:

- GapStat_Table_[Date]_[Time].txt : comma delimited tables with columns holding the *k*-values, gap statistic values, and standard errors, which collectively represent gap statistic analysis results.
- GapStatClustering_Figure_[Data]_[Time].fig : A MATLAB figure, containing the gap statistic analysis results shown in two plots: the gap statistic plot (left panel) used to determine the optimal *k*-value to cluster the input data, and the clustering results (right panel) when using the optimal *k*-value, which is also reported in the plot title. The clustering plots appear as 2-dimensional (height and time), but can be rotated using the rotation tool in MATLAB to view in 3-dimensional

space (slope, height, and time). A static image file in PNG format of this figure is also saved with the same filename.

- 3DRawData_Figure_[Data]_[Time].fig : A MATLAB figure, containing the clustering results when using the optimal *k*-value in the raw 3-dimensional space defined by slope, height, and time. This is a standalone figure of the right panel plot displayed in "GapStatClustering_Figure". The clustering plots can be rotated using the rotation tool in MATLAB to view in 3-dimensional space (slope, height, and time). A static image file in PNG format of this figure is also saved with the same filename.
- 3DStandardizedData_Figure_[Data]_[Time].fig : A MATLAB figure, containing the clustering results when using the optimal *k*-value in the log-transformed and then standardized 3-dimensional space defined by slope, height, and time. The clustering plots can be rotated using the rotation tool in MATLAB to view in 3-dimensional space (slope, height, and time). A static image file in PNG format of this figure is also saved with the same filename.

Note that the [DATE] and [TIME] placeholders in the filenames represent the date and time at the time that STADIA was run to produce that content. As noted above, each figure is saved as a static PNG image, while the user can interact with the MATLAB figure version of each file.

BIBLIOGRAPHY

Mahserejian, Shant M., Jared P. Scripture, Ava J. Mauro, Elizabeth J. Lawrence, Erin M. Jonasson, Kris S. Murray, Jun Li, Melissa Gardner, Mark Alber, Marija Zanic, Holly V. Goodson. "Quantification of Microtubule Stutters: Dynamic Instability Behaviors That Are Strongly Associated with Catastrophe."

Tibshirani, Robert, Guenther Walther, and Trevor Hastie. 2001. "Estimating the Number of Clusters in a Data Set via the Gap Statistic." *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 63 (2): 411–23. https://doi.org/10.1111/1467-9868.00293.