AFFECT MATLAB Toolbox for Clustering Dynamic Data (Version 0.5)

Kevin S. Xu, Mark Kliger, and Alfred O. Hero III
University of Michigan

February 20, 2012

The AFFECT (Adaptive Forgetting Factor for Evolutionary Clustering and Tracking) MATLAB toolbox is designed for clustering dynamic data sets and tracking communities in dynamic networks. For a description of the adaptive evolutionary clustering framework and algorithms, please see the paper that accompanies the toolbox (Xu, Kliger, & Hero III, 2011).

The AFFECT MATLAB toolbox currently includes the following:

- Batch implementations of AFFECT with k-means, hierarchical, and spectral clustering
- Demos of AFFECT with k-means, hierarchical, and spectral clustering on two colliding Gaussians simulated data
- Demo of AFFECT with spectral clustering on MIT Reality Mining data

Usage

All of the batch AFFECT clustering functions require at least three inputs, where T denotes the number of time steps in the data:

- `ids`: Length T cell array corresponding to the IDs (names) of the objects corresponding to the rows and columns of the proximity matrices stored in the cell array `W`
- `W`: Length T cell array corresponding to the proximity (similarity or dissimilarity, depending on the clustering method) matrices of active objects at each time step
- `num_clust`: Can be one of three options
    - A scalar specifying a fixed number of clusters over all time steps
    - A length T vector specifying the desired number of clusters at each time step
    - A string specifying the name of a heuristic for choosing the number of clusters, e.g. `'silhouette'` specifies to maximize the average silhouette width

The type of proximity matrix differs by clustering method:

- k-means: matrix of all pairs of dot products between objects
- Agglomerative hierarchical clustering: any type of dissimilarity matrix

- Spectral clustering: any type of similarity or graph adjacency matrix

The syntax and a list of options for each function can be found by typing `help function_name` at the MATLAB command line, where `function_name` is the name of the function.

Recommended MATLAB toolboxes

Some of the functionality in the AFFECT MATLAB toolbox depends on code from other MATLAB toolboxes. The dependencies are described in the following.

Statistics Toolbox

Required for the following functionality:

- To use AFFECT with agglomerative hierarchical clustering (requires functions `linkage` and `cluster`)
- To run the two Gaussians demo of AFFECT with hierarchical and spectral clustering (requires functions `pdist` and `squareform`)
- To use k-means to discretize the eigenvectors of the adjacency matrices for spectral clustering (set optional input `disc_type` in `batch_affect_spectral` to `'ortho'` to use an alternate discretization method that does not require the Statistics Toolbox)

Bioinformatics Toolbox

Required for the following functionality:

- To pre-process adjacency matrices for spectral clustering by first removing connected components (using the `remove_cc` optional input in `batch_affect_spectral`)
- To reorder rows of the cluster heat map by hierarchical clustering (optional input in `clu_heatmap`)

Description of demos

The demos correspond to two experiments from the paper (Xu, Kliger, & Hero III, 2011). Please consult the paper for more details about the experimental setup.

Two colliding Gaussians (k-means, hierarchical and spectral clustering)

The setup of this experiment is illustrated in Figure 1. 40 samples are drawn from a mixture of two 2-D spherical Gaussian distributions with mixture proportion 1/2. The mean of the red component is gradually moved toward that of the blue component, resulting in significant overlap between samples from different components. The mixture proportion is then changed to 3/8 and 1/4 to simulate objects changing clusters. At each time step, we draw a new independent sample.

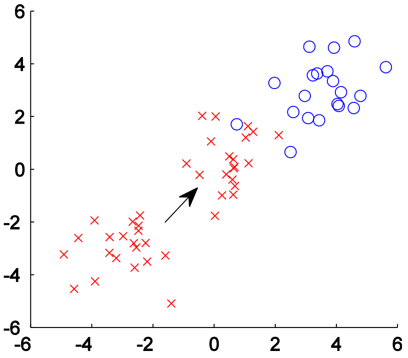

Figure 1. Setup of two colliding Gaussians experiment: one mixture component is slowly moved toward the other, then a change in component memberships is simulated.

MIT Reality Mining (spectral clustering)

This experiment involves a real data set with objects entering and leaving at different time steps. The data was collected by recording cell phone activity of 94 students and staff at MIT over a year (Eagle, Pentland, & Lazer, 2009). Each phone recorded the Media Access Control (MAC) addresses of nearby Bluetooth devices at five-minute intervals. Using this device proximity data, we construct graph adjacency matrices where the edge weight between two participants corresponds to the number of intervals where they were in physical proximity during each week.

Description of files

The following files are all a user should need to interact with. Descriptions of the remaining files can be found in the appendix.

- `batch_affect_kmeans.m`: Perform AFFECT k-means clustering in batch mode on a sequence of matrices consisting of all pairs of dot products between objects
- `batch_affect_linkage.m`: Perform AFFECT agglomerative hierarchical clustering in batch mode on a sequence of dissimilarity matrices
- `batch_affect_spectral.m`: Perform AFFECT spectral clustering in batch mode on a sequence of similarity or adjacency matrices
- `Demo_kmeans_Gaussians.m`: Demo script that runs AFFECT k-means clustering on the two colliding Gaussians simulated data
- `Demo_linkage_Gaussians.m`: Demo script that runs AFFECT agglomerative hierarchical clustering on the two colliding Gaussians simulated data
- `Demo_spectral_Gaussians.m`: Demo script that runs AFFECT spectral clustering on the two colliding Gaussians simulated data
- `Demo_spectral_reality.m`: Demo script that runs AFFECT spectral clustering on the MIT Reality Mining data and displays a heat map of the clustering results
- `clu_heatmap.m`: Create a heat map of the cluster evolution over time
- `reality.mat`: MIT Reality Mining data used in `Demo_spectral_reality.m`

Changelog

Version 0.5

- Added batch implementations of AFFECT with k-means and hierarchical clustering
- Changed data structure for proximity matrices; object IDs (names) are now kept in a separate cell array from the proximity matrices
- Corrected an object ordering bug that produced incorrectly ordered clustering results when object IDs were not sorted
- Corrected a bug in `permute_clusters_opt.m` that matched clusters suboptimally when the number of clusters changed over time
- Added demos of AFFECT k-means, hierarchical, and spectral clustering on two colliding Gaussians simulated data
- Updated demo of AFFECT on MIT Reality Mining data
- Significantly expanded Readme file

Version 0.2

- Fixed a bug that prevented the modularity and silhouette cluster selection heuristics from being used
- Created two options for eigengap heuristic: one that prompts the user to pick the number of clusters only once (the number is kept constant over all time steps) and one that prompts to pick the number of clusters at each iteration in each time step (annoying but allows the number of clusters to vary over time)
- Added functions `modularity.m` and `parse_inputs.m` that were mistakenly not included in version 0.1
- Added recommended MATLAB toolboxes and descriptions of files to the Readme

Version 0.1

- Initial release containing batch implementation of AFFECT with spectral clustering and demo on MIT Reality Mining data

Appendix: Description of additional files

Most users should not need to interact with the following files. Unless otherwise noted, all files were written by the authors of this toolbox.

- `estimate_alpha.m`: Used for estimation of the forgetting factor in the AFFECT procedure
- `clu_sample_stats.m`: Compute sample statistics over each block of the adjacency matrix in order to estimate the forgetting factor
- `permute_clusters_opt.m`: Match clusters between time steps by enumerating all permutations to find the optimal one (not recommended for more than four clusters)
- `permute_clusters_greedy.m`: Match clusters between time steps in a greedy fashion

- `kmeans_sim.m`: Perform static k-means clustering on a similarity matrix consisting of all pairs of dot products between objects (used for static clustering step in `batch_affect_kmeans.m`)
- `spectral_cluster.m`: Perform static spectral clustering on a similarity or adjacency matrix (used for static clustering step in `batch_affect_spectral.m`)
- `clumat2cluvect.m`: Converts binary matrix representation of clustering result to vector representation
- `select_clu_modularity.m`: Used to select the number of clusters with highest the modularity quality function
- `select_clu_silhouette.m`: Used to select the number of clusters to minimize the average silhouette width, another quality function
- `cell2matseq.m`: Convert a cell array storing the sequence of adjacency matrices into a 3-D matrix containing the union or intersection of active nodes
- `modularity.m`: Calculates the modularity of a clustering result (used to choose the number of clusters in `select_clu_modularity.m`)
- `generate_Gaussians_data.m`: Generate simulated data for two colliding Gaussians experiment

`discretisation.m` and `discretisationEigenVectorData.m` are redistributed without modification from the MATLAB normalized cut spectral clustering toolbox (Shi, 2004).

`valid_RandIndex.m` is redistributed without modification from Wang's (2009) MATLAB toolbox for estimating the number of clusters.

References

Eagle, N., Pentland, A., & Lazer, D. (2009). Inferring friendship network structure by using mobile phone data. *Proceedings of the National Academy of Sciences, 106*(36), 15274–15278.

Shi, J. (2004). *MATLAB Normalized Cuts Segmentation Code*. Retrieved from http://www.cis.upenn.edu/~jshi/software/

Wang, K. (2009). *Toolbox for estimating the number of clusters*. Retrieved from http://www.mathworks.com/matlabcentral/fileexchange/13916-simple-tool-for-estimating-the-number-of-clusters

Xu, K. S., Kliger, M., & Hero III, A. O. (2011). Adaptive evolutionary clustering. Retrieved from http://arxiv.org/abs/1104.1990