**K-Means Clustering EViews add-in**

Erhard Menker

June 6, 2017

**Overview**

This document explains the theory & motivation behind k-means clustering, and presents the application of the algorithm using the EViews add-in. The algorithm’s implementation adheres to that given by Dr. Andrew Ng’s machine learning course.[[1]](#footnote-1) However, any deviations from the pseudocode are of course the error of the add-in’s author. Inquiries of any kind are warmly welcomed; the add-in’s author, Erhard Menker, can be reached via email ([ejmenker@gmail.com](mailto:ejmenker@gmail.com)), and the working version of the project will continue to be developed on Github.[[2]](#footnote-2)

**Cluster Analysis Motivation**

Cluster analysis is an unsupervised machine learning algorithm, meaning it is applied to “unlabeled data” (classification of observations is not given in the dataset). Given a dataset of n observations, k-means clustering assigns each observation in the dataset to belong to one of cluster centroid 1 or 2 or … or k, conditioned that the # of centroids is less than n. A cluster’s centroid is the arithmetic average of each series for its associated observations. This assignment is done by finding the centroids that minimizes the cost function, a measure that values centroids that have close proximity to their associated points. Therefore, clustering algorithms are a useful form of exploratory data analysis and can be used in EViews with time series data (e.g. classifying macroeconomic regimes over a country’s history) & cross section data (e.g. customer segmentation).

**User Arguments**

* *Mandatory*
  + **k** – the # of centroids that the dataset with n observations will be partitioned into (constrained to be less than n)
* *Optional* 
  + **quiet** – shut off the add-in’s log messages
    - defaults to presenting log messages
  + **inits** – the # of random initializations & solves of the cluster centroids to occur
    - defaults to 3
  + **max\_iters** – the maximum # of times the cluster centroids are allowed to move based on its associated observations for a given solve of centroids
    - defaults to 10
    - if set to “NONE”, will continue until convergence occurs
  + **series** – a space delimited string of the series on the workfile page to be included in the cluster analysis
    - defaults to including all series on the workfile page
  + **smpl** – observations to be included in the clustering algorithm
    - defaults to the sample at time of function call
    - if smpl = @all or smpl = all, then set sample to equal the range
  + **impute** | **interpolate** – standalone argument that interpolates a series’ missing values
    - defaults to not interpolating
    - for time series pages, linearly interpolate series
    - for unstructured workfiles, impute with the series’ median

**K-Means Algorithm Summary[[3]](#footnote-3)**

1. each series is normalized to remove scaling impacts
2. imputation occurs if the user selects this option
3. any observation with an NA in at least 1 series is removed from the clustering process (an NA is returned for that period in the output)
4. for each random initialization to be solved:
   * initialize k random observations as the cluster centroids
   * while the cluster centroids continue to converge to the solution OR the maximum # of cluster moves is NOT reached for the random initialization:
     + find the closest centroid that each observation corresponds to
     + recalculate the cluster centroids as the arithmetic mean of its associated observations
   * find the closest centroid that each observation corresponds to
   * calculate the cost function (a measure of how distant all the observations are from their associated centroid)
   * if the cost function is the smallest for all initializations thus far:
     + store the centroids & associated observations as the optimal clustering thus far

**Add-in Output**

* 2 objects are returned to the add-in page:
  + **obs\_cluster** – a series object where each observation states the cluster # to which the observation belongs
  + **kmeans\_results** – a text object declaring, for each centroid, how the mean of each of its series (both nominal & percentage differences) compares to the overall series mean

**Example Application**

***Program:***

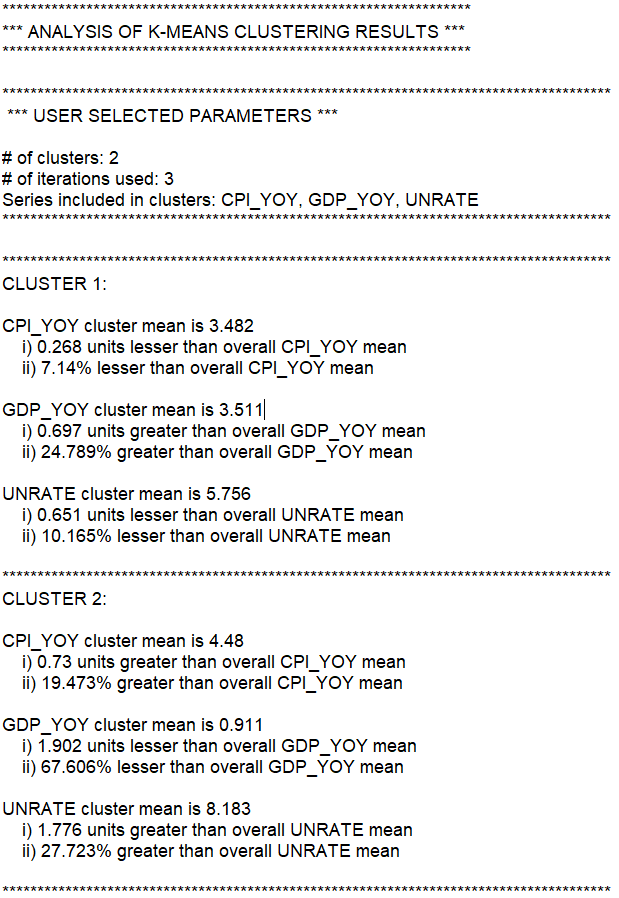
|  |
| --- |
| ' create an annual workfile  wfcreate(wf=k\_means\_example, page=DATA\_A) a 1975 2016  ' fetch series (accessed via FRED API)  ' a) unemployment rate  fetch fred::unrate  ' b) gdp yoy  fetch fred::a191ro1q156nbea  rename a191ro1q156nbea gdp\_yoy  ' c) cpi yoy  fetch fred::cpiaucns  series cpi\_yoy = @pcy(cpiaucns)  delete cpiaucns  ' calculate 2 cluster centroids between 1975 & 2016  exec .\..\kmeans.prg(k = 2) |

***Output:***

a. series object (*obs\_cluster*) indicating which observations correspond to the 2 centroids



b. text file (*kmeans\_results01*) stating how mean of each series for each centroid compares to the series’ aggregate mean



***Analysis:***

This k-means application looks at annual U.S. macroeconomic data between 1975 & 2016. Level unemployment & year-over-year percentage changes in real GDP & the unemployment rate capture the state of the US macroeconomy over the past few decades. Choosing 2 clusters centroids means that there are going to be 2 states of the economy captured. Looking at the text file, we can see there is cluster 1 (higher GDP growth and lower unemployment level & CPI growth) and cluster 2 (lower GDP growth and higher unemployment level & CPI growth).

These economic regimes correspond to understood macroeconomic U.S. history; observations from cluster 2 appear during periods of U.S. recession including in the early 1980s & 1990s, the bursting of the tech bubble, and the financial crisis (with more 2s in the financial crisis than the rest of the recessions here combined, highlighting the severity & slow recovery).

This serves as a vindication of accuracy; more interesting applications require increased sophistication of problem setup. For example, the economist could consider adding more macroeconomic series & increasing k to detect for more complicated regimes (e.g. high GDP growth, lower unemployment & higher inflation). Recessions are defined by GDP; another application would be to see if different economic indicators, like manufacturing/trade data, could be used to pick up recessions (or even serve as leading indicators of recessions when the series are considered together).

To do analysis on 1 centroid of the data, conditional sample definitions can be made:

|  |
| --- |
| ' constrain the sample to the recession values  smpl @all if obs\_cluster = 2 |

1. <https://www.coursera.org/learn/machine-learning/lecture/93VPG/k-means-algorithm> [↑](#footnote-ref-1)
2. <https://github.com/ErhardMenker/kMeans4EViews> [↑](#footnote-ref-2)
3. For a more detailed explanation with visualizations, see: <http://cs229.stanford.edu/notes/cs229-notes7a.pdf> [↑](#footnote-ref-3)