EIGENVECTOR RESEARCH INCORPORATED
Research, Training, and Software

196 Hyacinth Road
Manson, WA 98831
www.Eigenvector.com

Selection of Representative Learning and Test Sets Using the Onion Method

Neal B. Gallagher, Donal O'Sullivan

Key words: FTIR, Extended Multiplicative Inverse Scatter Correction, Extended Multiplicative Scatter Correction

Introduction: Often it is of interest to split a single data set into two: one for calibrating a model (a learning set) and one for testing a model's performance (a validation set). Two approaches used by the `splitcaltest` function in Solo and PLS_Toolbox (www.eigenvector.com) to split a single data set are the well-known Kennard-Stone algorithm[1] and a less well-known "onion" algorithm. This white paper provides a brief comparison of the two methods.

Kennard-Stone: The Kennard-Stone (KS) algorithm starts by finding two samples in that have the largest Euclidean distance; these are the two samples furthest apart. Subsequent points are selected that maximize the Euclidean distance from previously selected points. The result is that, for a given fraction (*fraction*) of the data to be parsed into the calibration set, samples are selected to smoothly fill the data space. The remaining samples are placed into the test set. An example is shown in Figure 1 for a synthetic data set that has 1000 samples and 2 variables with a multivariate normal distribution. In this case, 66% of the samples were placed into the calibration set. As expected, the test set is interior to the exterior points of the calibration set. This splitting therefore avoids extrapolation when a model is applied to the test set.

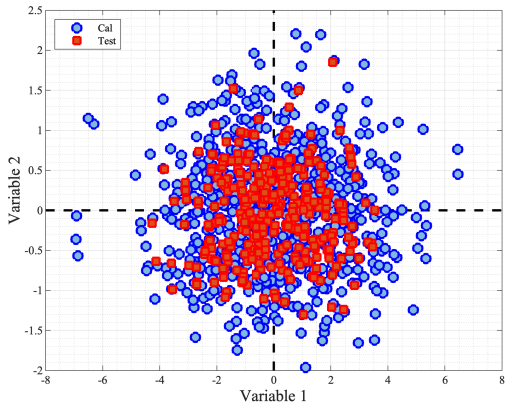

Figure 1: Samples selected using the Kennard-Stone algorithm: calibration set (blue circles) and test set (red squares).

Onion: The onion algorithm selects samples in layers akin to the structure of an onion. The external layer is placed into the calibration set and the number of points in this layer is dictated by the '*loop fraction*'. For example, for a data set with *M* samples, the number in the external layer is (*loop fraction*)*(*fraction*)**M*. A small 'loop fraction' means that more test samples are closer to the outside of the data space. The next layer has the same number of samples (if sufficient samples are present) and are assigned to the test set. The process is repeated *nonion* times where *nonion* is a small integer corresponding to the number of layers. The remaining samples in the "onion core" are then randomly assigned to calibration and test sets.

For the same data set shown in Figure 1, Figure 2 shows the results for the onion algorithm with a *loopfraction* = 0.1. Comparison of Figures 1 and 2 show more test samples closer to the exterior of the data space for the onion method.

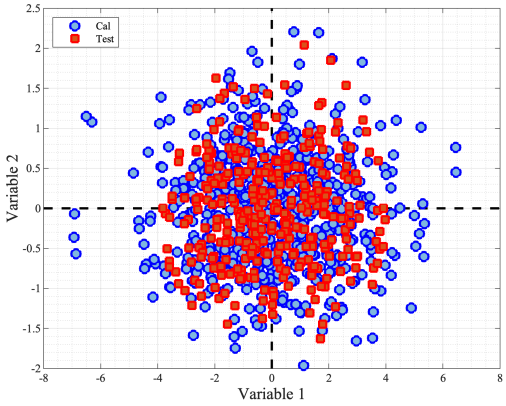

Figure 2: Samples selected using the Onion algorithm: calibration set (blue circles) and test set (red squares). [*fraction* = 0.66, *nonion* = 3, *loopfraction* = 0.1]

The onion method uses a Euclidean distance and selects samples using the "distslct" algorithm described in Reference [2]. The distslct algorithm was originally designed to initialize a D-optimal algorithm[3] and can be considered a sample selection criteria based on a D-optimal criterion. Figure 3 shows the results for the onion method when *loopfraction* = 0.5. Increasing *loopfraction* moves the test samples closer to the interior of the data space and now the onion layers are evident. However, the test samples appear to be biased along the axis for Variable 2. Changing the distance measure from Euclidean to Mahalanobis removes the bias as shown in Figure 4.

196 Hyacinth Road
Manson, WA 98831
www.Eigenvector.com

Discussion: The Kennard-Stone algorithm is easy to use and translates well to smaller data sets. In contrast, implementation of the onion algorithm requires more thought towards using an appropriate *loopfraction* as the data set gets smaller; it makes sense to use a larger *loopfraction* for smaller data sets (~M<30). It is

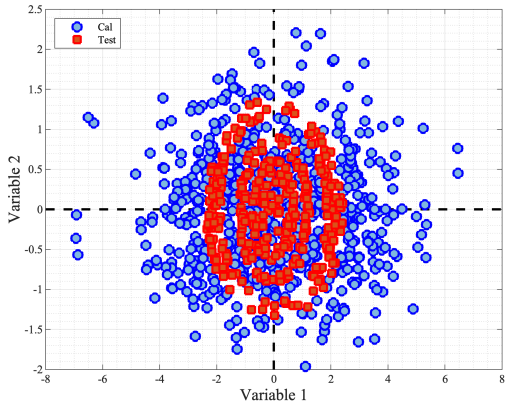

Figure 3: Samples selected using the Onion algorithm: calibration set (blue circles) and test set (red squares). [*fraction* = 0.66, *nonion* = 3, *loopfraction* = 0.5]

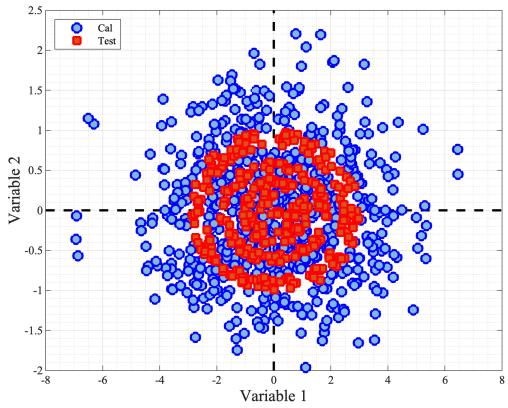

Figure 4: Samples selected using the Onion algorithm with a Mahalanobis distance measure: calibration set (blue circles) and test set (red squares). [*fraction* = 0.66, *nonion* = 3, *loopfraction* = 0.5].

expected that the iterative approach used by the onion method could be easily extended to the KS algorithm. Additionally, the Mahalanobis distance can also be used with the KS algorithm (Figure 5).[4] Reference [4] has a nice discussion of other methods for splitting data sets into calibration and test sets.

Computation time for large data sets with many samples can get very large for both algorithms discussed. For example, Figure 6 shows execution time for a data set with M = 500 to 6000 samples and 10 variables with KS taking significantly longer: At 6000 samples the KS algorithm took 107 s while onion took <1 s. In any case, it is recommended that scores (e.g., from principal components analysis or partial least squares) be used to reduce the number of variables significantly.

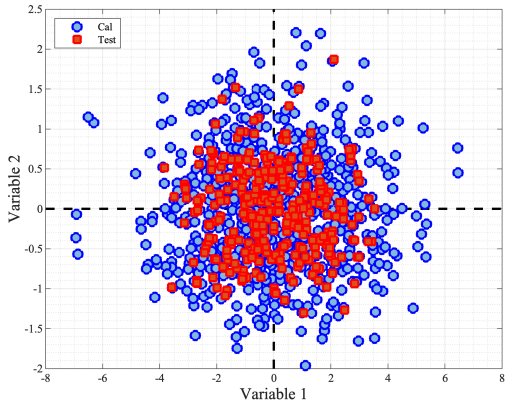

Figure 5: Samples selected using the Kennard-Stone algorithm with a Mahalanobis distance measure: calibration set (blue circles) and test set (red squares).

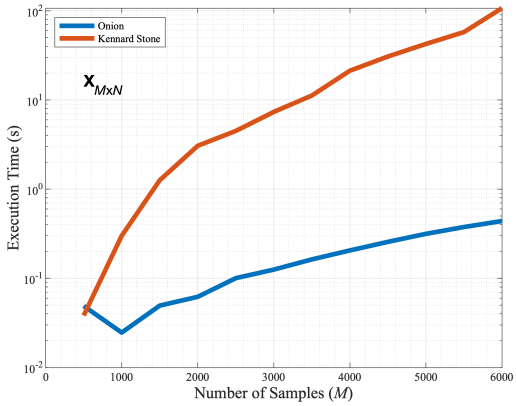

Figure 6: Execution time for the Onion (blue) and Kennard-Stone (red) algorithms versus number of samples. [onion settings: *fraction* = 0.66, *nonion* = 1, *loopfraction* = 0.5]

References:

[1] Kennard, RW, Stone LA, "Computer Aided Design of Experiments, *Technometrics*, **11**(1), 137-148 (1969).

[2] Gallagher, NB, Shaver, JM, Martin, EB, Morris, J, Wise, BM, Windig, W, "Curve resolution for images with applications to TOF-SIMS and Raman," *Chemo. and Intell. Lab. Sys.*, 2003, **77**(1), 105-117.

[3] de Aguiar, PF, Bourguignon, B, Khots, MS, Massart, DL, Phan-Than-Luu, R, "D-optimal designs", *Chemo. Intell. Lab. Sys.*, **30**, 199-210 (1995).

[4] Saptoro, A, Tadé, MO, Vuthaluru, H, "A Modified Kennard-Stone Algorithm for Optimal Division of Data for Developing Artificial Neural Network Models," *Chemical Product and Process Modeling*: **7**(1), Article 13 (2012), DOI: 10.1515/1934-2659.1645