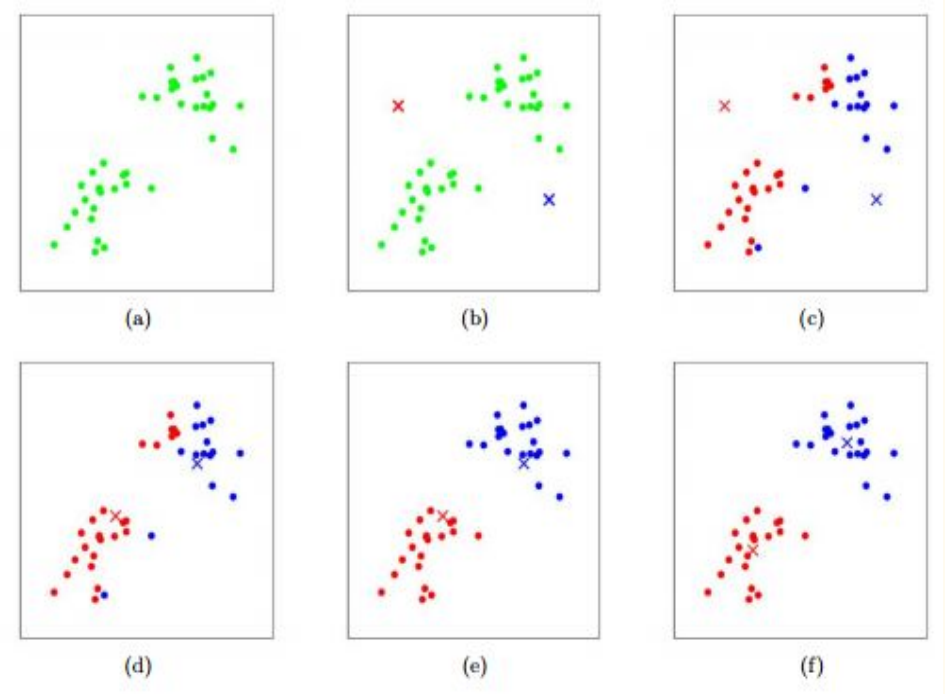
# K-means Clustering Algorithm

—

Andreas Francisco
Tu Timmy Hoang

# Algorithm

1. Place K points into the space represented by the objects that are being clustered. These points represent initial group centroids.
2. Assign each object to the group that has the closest centroid.
3. When all objects have been assigned, recalculate the positions of the K centroids.
4. Repeat Steps 2 and 3 until the centroids no longer move. This produces a separation of the objects into groups from which the metric to be minimized can be calculated.

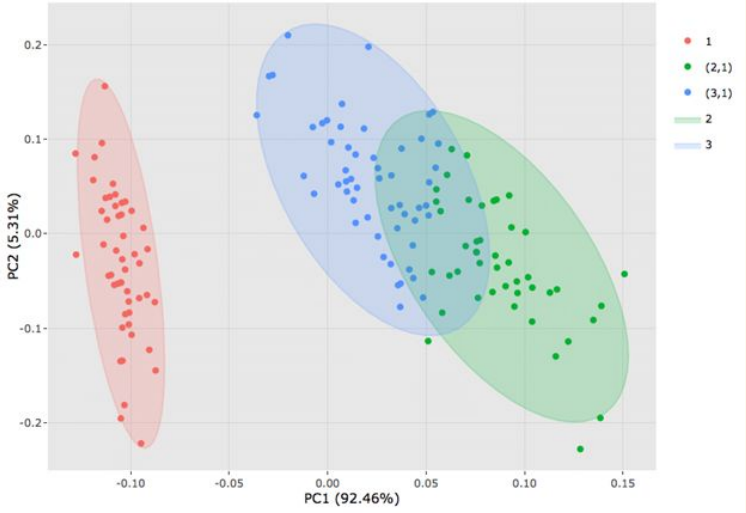


# Parallelization

- Parallelize each distance from a centroid using openACC or openMPI.
- Reducing data computation by splitting data.
- Possibly using buffers to deal with edge cases in data division.

# Fuzzy Algorithm

- Now we want to partition the points into Cj sets but points can be in more than one set so the representation of each point is a sum of the likelihood a point belongs to each cluster Cj.
- Continue computing the new centroids until the coefficient between two iterations of a point being in a cluster Cj has not changed more than epsilon.
- Compute the final centroids of each cluster and return the solution.
- Parallelize it the same as regular Algorithm.



# Comparison

- Check speed of convergence for the serial implementation of Hard Clustering and Fuzzy Clustering.
- Compare the accuracy of the solutions between them using a message passing interface.