# Project Proposal

## Possibilistic C-Means

**Holt Skinner**

**Machine Learning & Pattern Recognition**

### Description

In the world of clustering algorithms, the K Means and Fuzzy C-Means Algorithms remain popular choices to determine clusters. The basic K Means clustering algorithm goes as follows.

```
1. Initialize K cluster centers (Random or specifically
chosen from data set)
2. Place all points into the cluster of the closest
prototype
3. Update memberships and cluster centers
4. Repeat until Clusters Stabilize or until a certain
number of iterations.
```

The Fuzzy C-Means Algorithm improves upon K Means by allowing data points to have a membership in more than one cluster, given as a number between 0-1. All of the membership values for a particular data point must add up to one. Possibilistic C Means (PCM) is an algorithm created by Dr. Jim Bezdek and Dr. Jim Keller that eliminates the probabilistic constraint that all membership values must add up to one. This allows points relatively far away from all of the clusters (outliers) to have negligible membership in all of the clusters. This is an important advantage because

noise data or incomplete data can be filtered out without altering the cluster centers. If the outliers were left in, it could drastically shift the cluster centers away from their true location. Despite its advantages, there is currently not an open source python library that supports the PCM algorithm. To solve this, the project will consist of an open source implementation of PCM in Python hosted on GitHub.

## Goals

### Short-Term

- Code Implementation
- Test with Well-Known Data Set (Iris Data)
- Display Results in a Scatter Chart

### Long-Term & Final

- Initizalize Clusters with K-Means and Fuzzy C-Means output.
- Run PCM on NFL Play Data.

## Benefits / Expected Outcomes

The ultimate goal for the project is to create a working implementation of the Possibilistic C-Means Algorithm that can be generalized for a multitude of use cases. This implementation will be released to Github to allow it for use by Data Scientists and Machine Learning Specialists, as well as a comparison point for my Senior Capstone Project that compares Machine Learning Models for predicting the outcomes of NFL Games.

## Algorithm

Let $X = \{x_1, x_2, \ldots, x_n\}$, where $x_k \in \Re^d$ is the set of vectors to be clustered.

```
Initialization: Set
        C, the number of clusters desired
        m, the fuzzifier
        ε, the convergence threshold
        V^(0) = {v_1^(0), ..., v_C^(0)} an initial set of cluster centers
//Note: The v_i^(0) can be chosen randomly from X or through
  other mechanisms//
Set t = 0
REPEAT
    DO FOR each k = 1, . . . , n
       IF d(x_k,v_i) = 0 for some subset of clusters, i.e., I_k ≠ φ
       THEN
         Set u_jk^(t) = 0 for j ∉ I_k and u_jk^(t) > 0 for j ∈ I_k, as in Eq. 8.3b
       ELSE
         Compute u_ik^(t) from Eq. 8.3a.
       ENDIF
    END FOR
    Set t ← t + 1
    Using U^(t-1), estimate V^(t) from Eq. 8.4.
UNTIL Σ_{i=1}^{C} ||v_i^(t) − v_i^(t-1)|| < ε
where ||*|| is any vector norm (like Euclidean).

//Note: There are other stopping criteria, including number
of iterations, but this is the most common.//
```

Where:

$$u_{ik} = \frac{1}{1 + \left(d^2(x_k, v_i)/\eta_i\right)^{1/(m-1)}}$$

and

$$\eta_i = \frac{\sum_{k=1}^{n} u_{ik}^m d^2(x_k, v_i)}{\sum_{k=1}^{n} u_{ik}^m} \quad \text{or} \quad \eta_i = \frac{\sum_{u_{ik}>\alpha} d^2(x_k, v_i)}{\sum_{u_{ik}>\alpha} 1}, \quad \text{for some } 0 < \alpha \leq 1$$

Calculated with the results from Fuzzy C-Means or set equal to 1

Source: Keller, Fundamentals of Computational Intelligence

## Libraries Utilized

- Numpy
    - Scientific Computing Python Library
- Matplotlib
    - Python Library for plotting graphs and charts
- Scikit Learn
    - Python Library with preloaded datasets and the Fuzzy C-Means Algorithm
    - Use for comparison and testing.

## Validation and Testing

For initial validation and testing, the Iris flower data set as collected by Ronald Fisher will be used, as it is easily available and the proper classiciation is already known. Data will be run through both the Project Built PCM and the Fuzzy C-Means as available in the Scikit-fuzzy Open Source Python Library. Since the data has been filtered for noise points, both algorithms should provide similar results. The labels in the data itself can also be used to verify.

https://en.wikipedia.org/wiki/Iris_flower_data_set

Once the algorithm is verified, it will then be tested on National Football League Play Data from 2009-2016 to assist in finding patterns in play styles. The dataset is openly available on Kaggle.com and is used in conjunction with a Senior Capstone Project shared with James Hurt and Payton Hosna.