A New Methodolgy for Unsupervised Competitive Learning

The Standard UCL Algorithm

The Problem Setup:

Given a set of N-dimensional examplar feature vectors $\{\boldsymbol{x}^{(q)}: q = 1,....,Q\}$, draw a set of M nodal vectors of N dimensions, $\{\boldsymbol{v}^{(m)}: m = 1,....,M\}$, and then train the nodal vectors by moving them into central more dense positions among the exemplar feature vectors so they are in positions to represent the classes of $\{\boldsymbol{x}^{(q)}: q = 1,....,Q\}$.

```
     x x   xx              x x x   x
  x x    x x x   x        x x xx x x x            x - feature vectors
   x   x   xx xx x    o   x x   x xx x
 o     x x   x x xx        x   x x   xx           o - initial nodal vectors
       x x   x x           x x   x x o
```

As we have seen, UCL algorithm selects each feature vector $\boldsymbol{x}^{(q)}$ at a time and finds the nodal vector $\boldsymbol{v}^{(m)}$ closest to it and then moves $\boldsymbol{v}^{(m)}$ toward $\boldsymbol{x}^{(q)}$. But this process can cause movement toward outliers, and adjust back and forth to learn and unlearn, so the process is not smooth and is not efficient.

Project Idea for Improving the Process

Do not move nodal vector toward each closest $\boldsymbol{x}^{(q)}$ but instead move it toward a center of the nearest k feature vectors. Here is one way the algorithm could work:

```
Step 1: for m = 1 to M do
            find k nearest feature vectors $\boldsymbol{x}^{(q1)}, ..., \boldsymbol{x}^{(qk)}$ closest to $\boldsymbol{v}^{(m)}$ and find their centroid $\mu$
            move $\boldsymbol{v}^{(m)}$ toward $\mu$ via
                $\boldsymbol{v}^{(m)} = \boldsymbol{v}^{(m)} + \alpha[\mu - \boldsymbol{v}^{(m)}]$     where $\alpha = 1 - (1/4)exp[-||\mu - \boldsymbol{v}^{(m)}||^2]$

Step 2: if any two nodal vectors are too close, replace them by their average vector
        if no nodal vector moves more than $\epsilon$ then stop
        else go to Step 1
```

The function α will go from near 1.0 to near 0.25 as $\boldsymbol{v}^{(m)}$ approaches μ. For this to work well it appears that we need extra nodal vectors because there will be more than a single centroid for each class of feature vectors.