DM - Assignment 04

Problem 1

1)

Formulating non-linear soft-margin SVM:

$$\min_{w,b,\xi} \frac{1}{2}\|w^2\| + C\sum_{i=1}^{N} \xi_i$$

SVM with ø(x):

$$s.t. \quad \forall\, i \in [1,N], \quad y_i(< w, \phi(x_i) > + b) \geq 1 - \xi_i;$$
$$\forall\, i \in [1,N], \quad \xi_i \geq 0.$$

Quadratic Programming & Dual Function:

$$\min_{\theta} \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j y_i\, y_j K(x_i, x_j) - \sum_{i=1}^{N} \alpha_i.$$

$$s.t. \sum_{i=1}^{N} \alpha_i y_i = 0\,, where\ \alpha_i \epsilon [0, C], i = 1,2, \dots, N.$$

$$f(x) = sign\left(\sum_{i=1}^{N} \alpha_i y_i \phi(x_i)\phi(x) + b\right) = sign\left(\sum_{i=1}^{N} \alpha_i y_i K(x_i, x) + b\right).$$

2)

1. Given $K_{i,j} = K(x_i,x_j)$, we only need to find a mapping ø(x):X->H, s.t. for any x,z belongs to X, k(x,z) = ø(x)•ø(z); saying one kernel matrix, we can find multiple such mappings out of it, and feature vectors w and bias b can thus be calculated with that.

2. No. Time and space complexity increases by calculating feature vectors comparing to merely calculate the kernel matrix.

Problem 2

1) Complexity of k-means & k-medoids (PAM)

Time requires compute the distance between two data points is linear to their dimensions, that is a complexity of `O(n)`. Saying the number of iterations of either algorithm is `i`.

K-means steps are like below:

1. Initialization: choose k random centroids from X. `-> O(k)`
2. Repeat the following steps:

    - Assignment: perform 1-NN classification to (re)assign each $x \in X$ to one of the cluster centroids. `-> O(Nn)`
    - Update: recompute each cj to be the centroid of all the points assigned to its cluster. `-> O(Nn)`

    Until the centroids/clusters stabilize (i.e., do not change)

So the total complexity of k-means is `O(iknN)`.

For each iteration, the PAM steps are like below:

1. Initialize: select k of the N data points as the medoids. `-> O(k)`
2. Associate each data point to the closest medoid. `-> O(Nkn)`
3. While the cost of the configuration decreases:

    - For each medoid m, for each non-medoid data point o: `-> multiply by k(N-k)`

        - Swap m and o, recompute the cost (sum of distances of points to their medoid) `-> O((N-k)n) cuz medoids need not to be computed`
        - If the total cost of the configuration increased in the previous step, undo the swap `-> O(C)`

So the total complexity of PAM is `i*(O(k) + O(Nkn) + k(N-k)*O((N-k)n)) = O(ikn(N-k)^2)`.

2) Shake & Bake Disimilarity

1. When $D(x,y) = L^{-1}\sum^{L}_{j=1}D_j(x,y) = 0$, because $D_j \in \{0,0.5,1\}$, so for every j, D_j has to be 0, in that condition x=y.
2. D(x,y) is clearly symmetrical regarding x and y, D(x,y) = D(y,x) stands.
3. Pick some $z \in X$, if z equals to x or y, then D(x,y) = D(x,z) + D(z,y); otherwise, for each j, we have the

following conditions:

C(x)=C(y)	C(y)=C(z)	C(x)=C(z)	D(x,y)	D(y,z)	D(x,z)
T	T	T	0.5	0.5	0.5
T	F	F	0.5	1	1
F	T	F	1	0.5	1
F	F	T	1	1	0.5
F	F	F	1	1	1

In all above cases, $D(x,y) \leq D(x,z) + D(y,z)$, so triangle rule stands.

Thus, by 1-3 we have proved shake & bake disimilarity is a distance metric.

3) Centroid by Attribute-wise Mode

The sum of absolute error (SAE) of a cluster w.r.t. mismatch distance d(x,y) = #{x[j] != y[j]} ($j \in [1,n]$ where n is the dimensionality of points) can be expressed as:

```
∑j∑i int(X(i,j) != C(j))
```

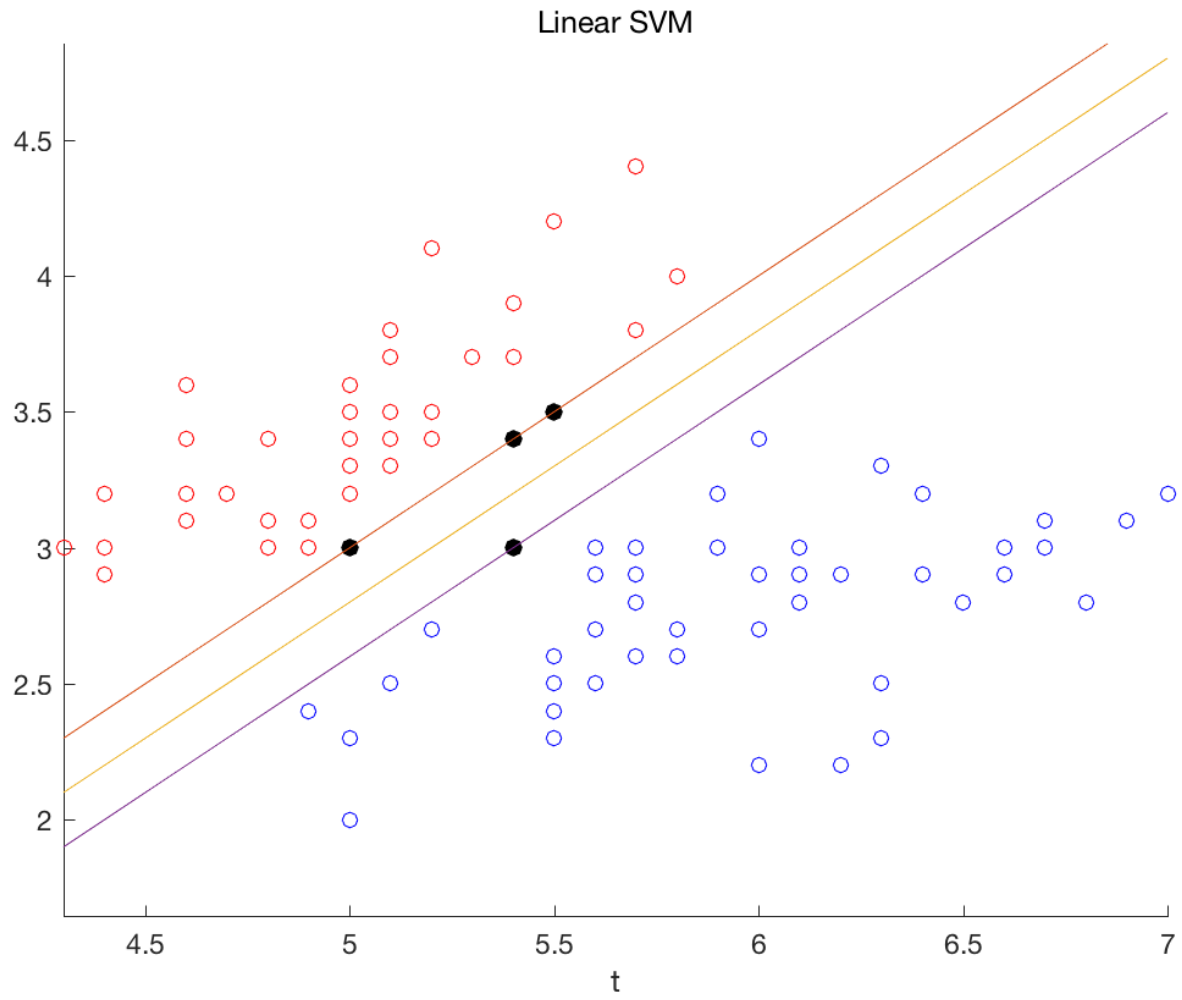
where $i \in [1,N]$, N is number of points in this cluster, and C is the centroid (nx1). It is clear that each attribute (j) here is independent, making it j different minimization problem, each one formulated as:

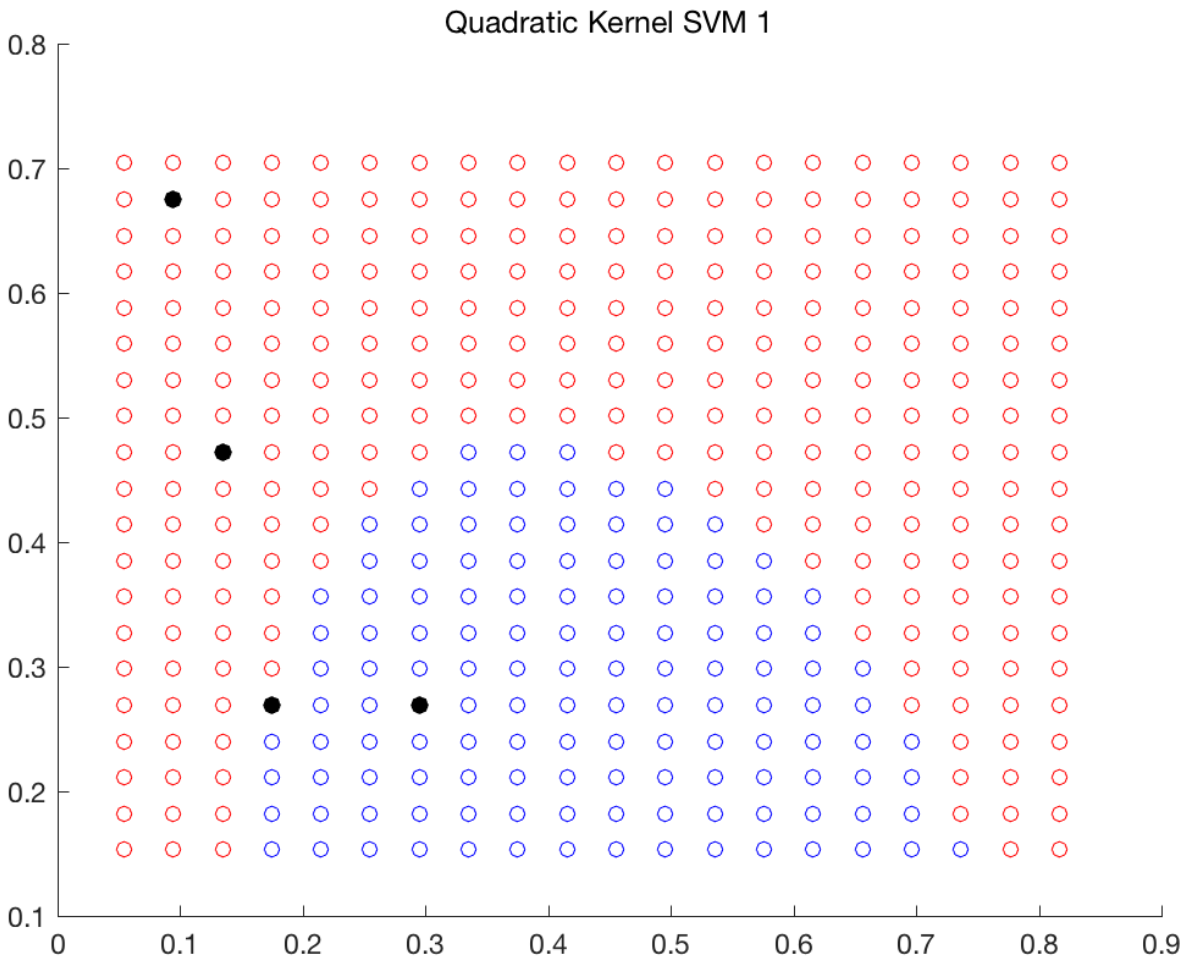
```
∑i int(X(i,j) != C(j))
```

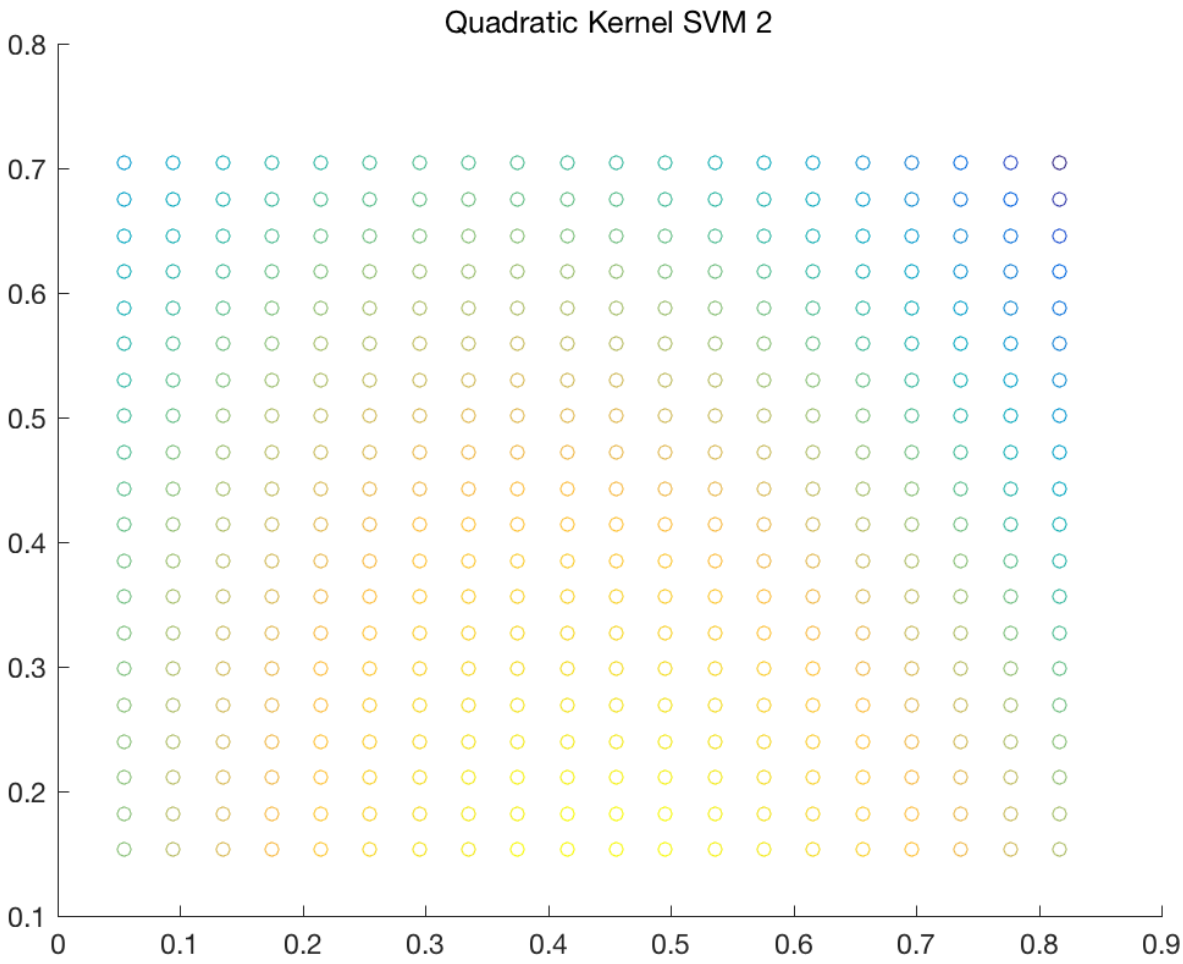
To make it smaller in each j, we should set as many X(i,j) equals to C(j), which means setting C(j) = mode(X(:,j)).

Thus the attribute-wise mode is indeed the centroid w.r.t the mismatch dissimilarity in terms of minimizing SAE in each cluster.

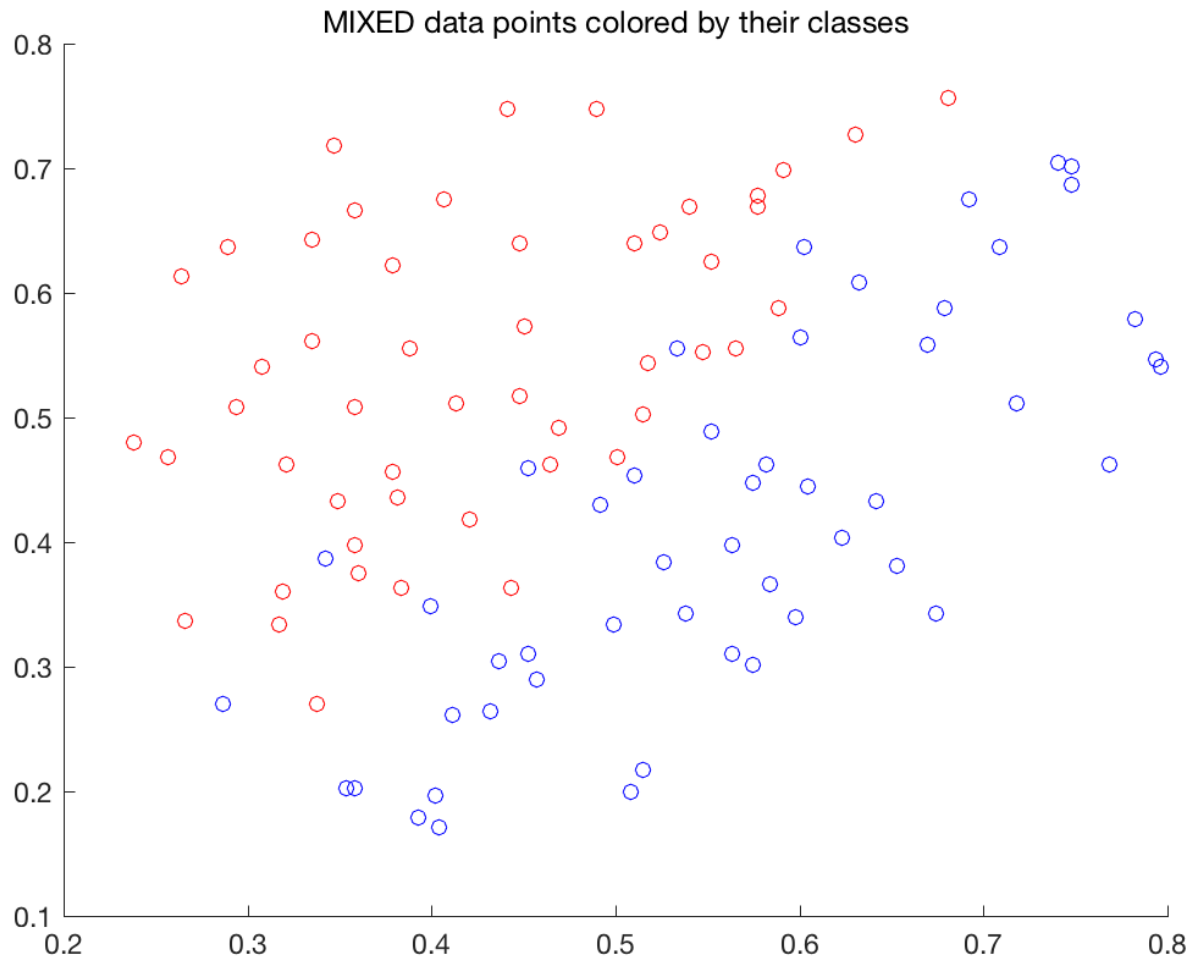
Problem 3

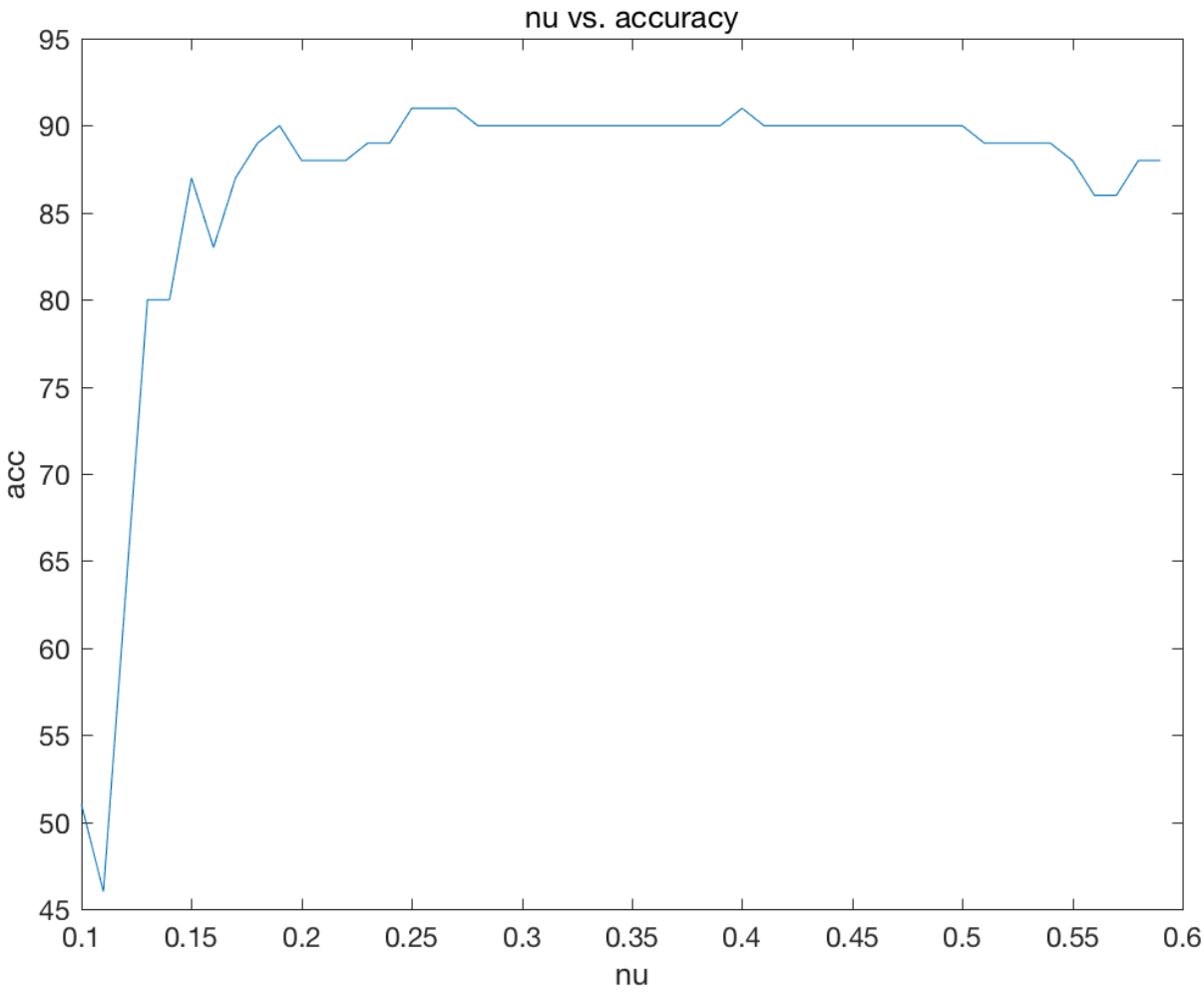

Linear SVM
4.5
4
3.5
3
2.5
2
4.5
5
5.5
6
6.5
7
t

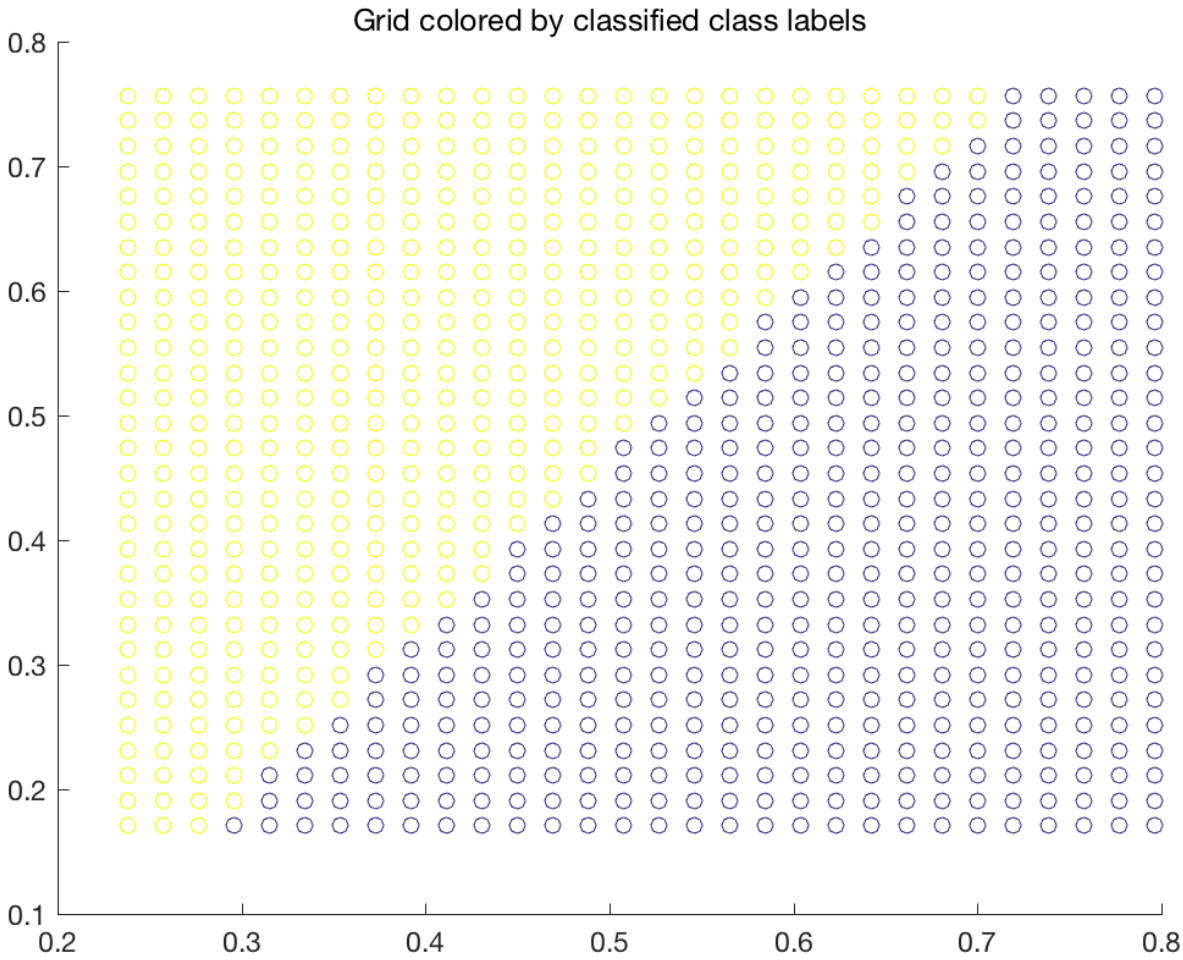

Quadratic Kernel SVM 1
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9


Quadratic Kernel SVM 2
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9

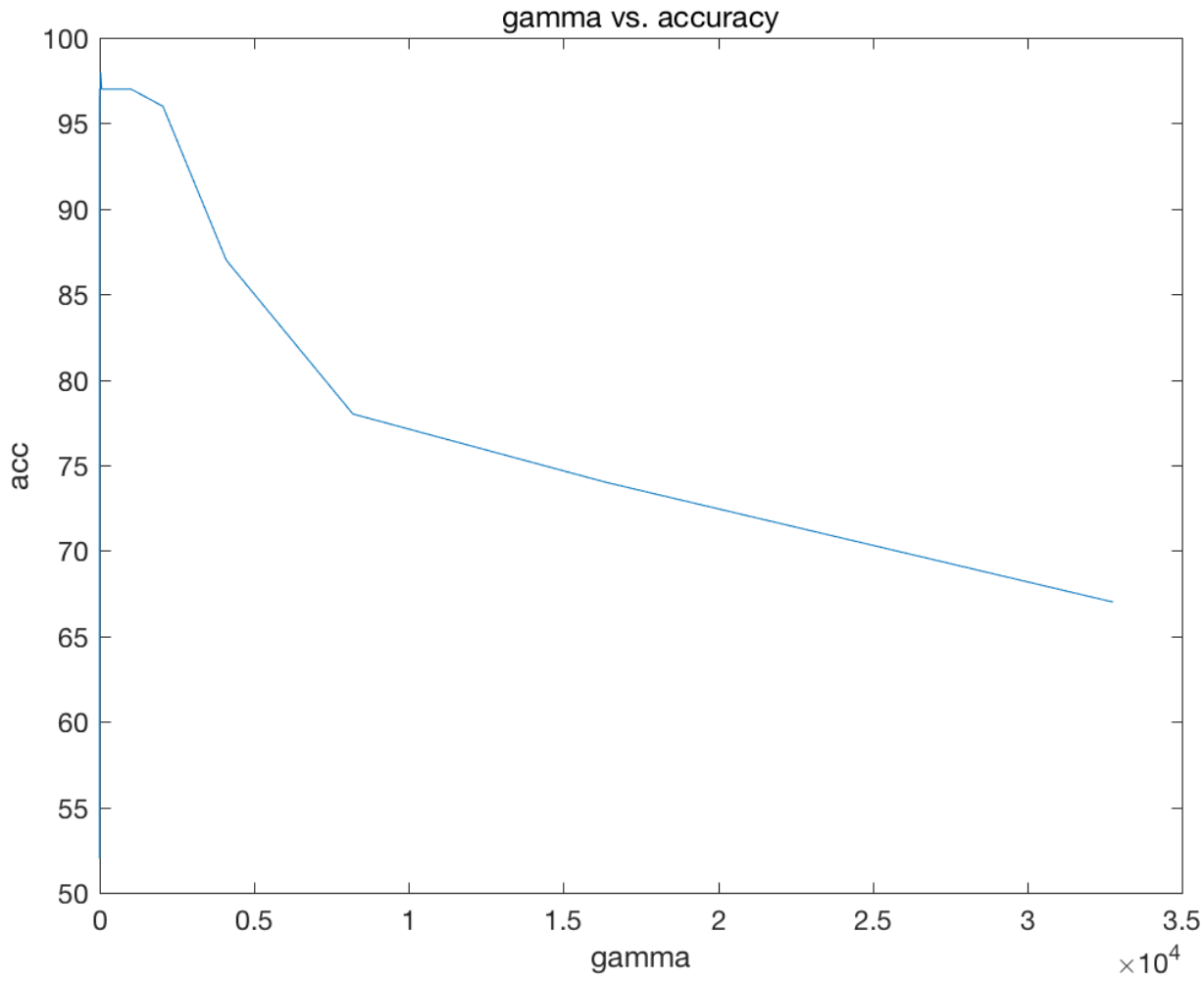
Problem 4


MIXED data points colored by their classes
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8


nu vs. accuracy
acc
nu
95
90
85
80
75
70
65
60
55
50
45
0.1
0.15
0.2
0.25
0.3
0.35
0.4
0.45
0.5
0.55
0.6

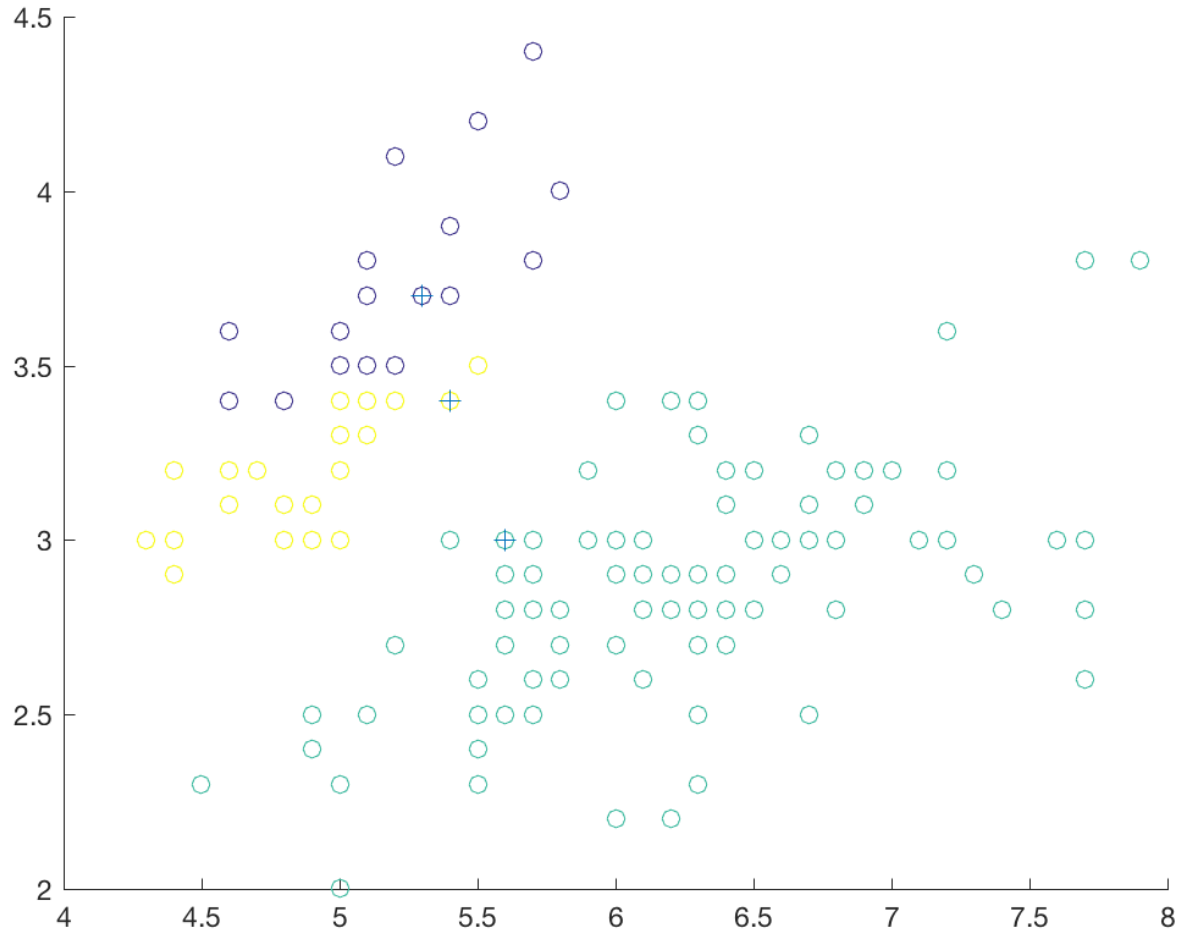

Grid colored by classified class labels
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8

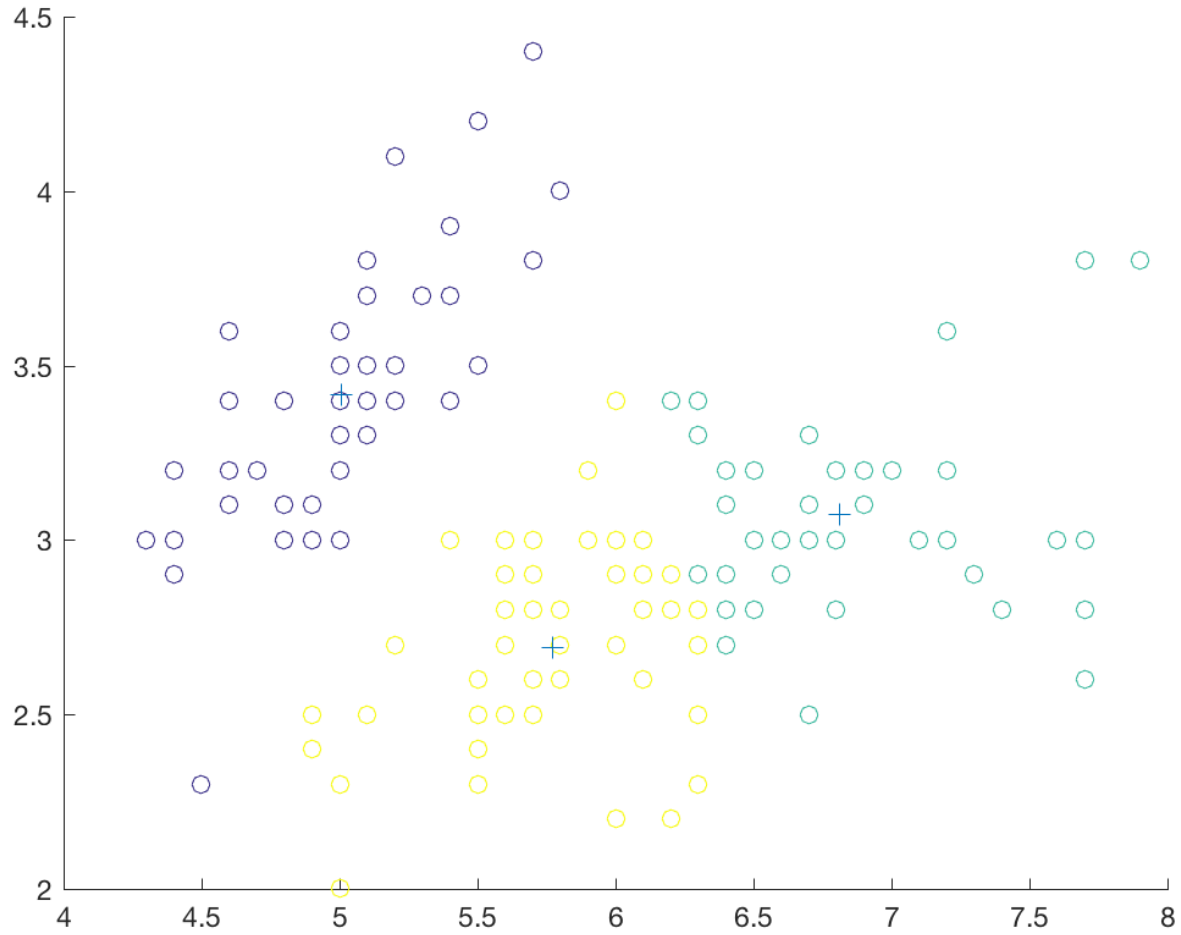

TARGET data points colored by their classes
0.8
0.75
0.7
0.65
0.6
0.55
0.5
0.45
0.4
0.35
0.3
0.25
0.3
0.35
0.4
0.45
0.5
0.55
0.6
0.65
0.7


gamma vs. accuracy
acc
gamma
×10^4
100
95
90
85
80
75
70
65
60
55
50
0
0.5
1
1.5
2
2.5
3
3.5

Problem 5

Illustrate clustering on a two-dimensional dataset

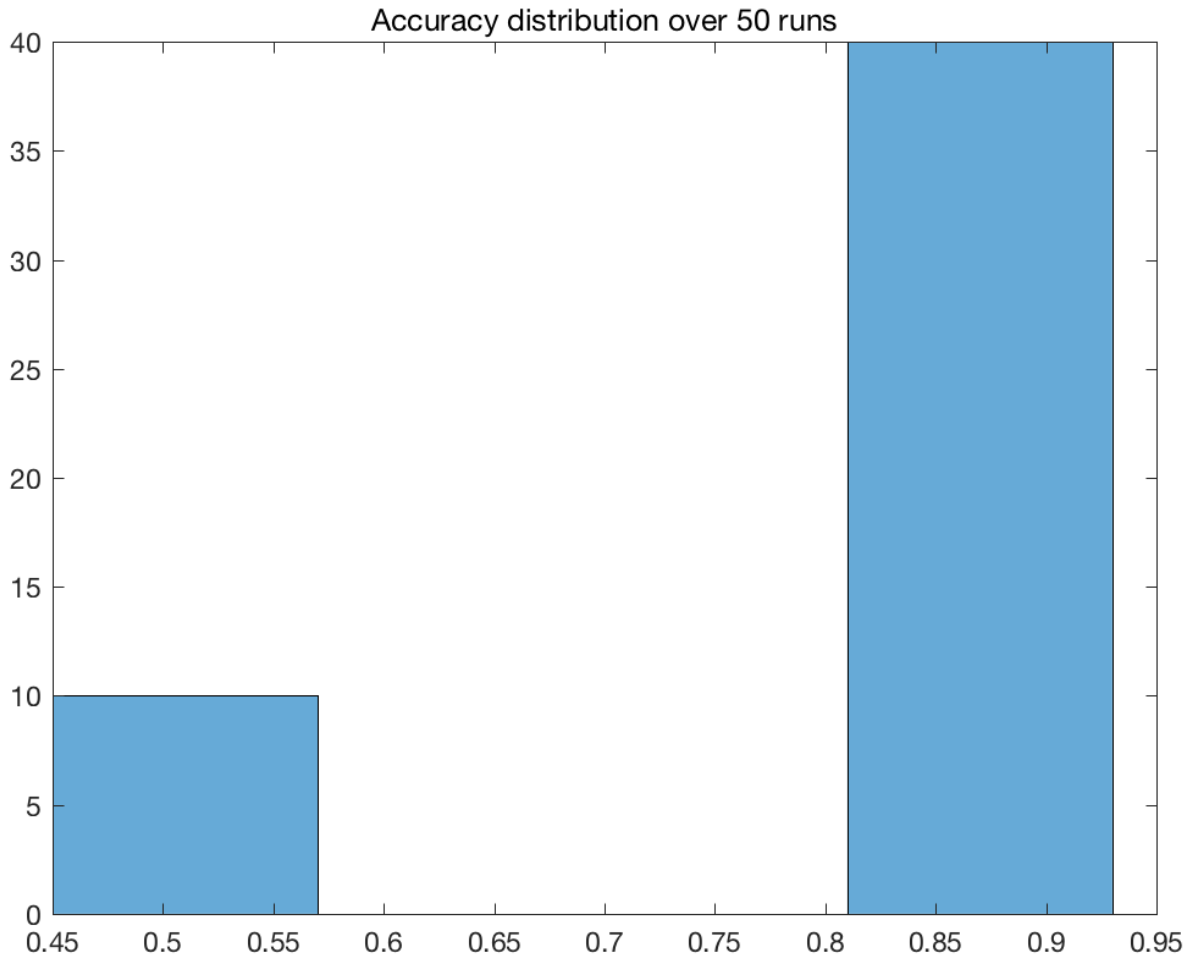

4.5
4
3.5
3
2.5
2
4
4.5
5
5.5
6
6.5
7
7.5
8

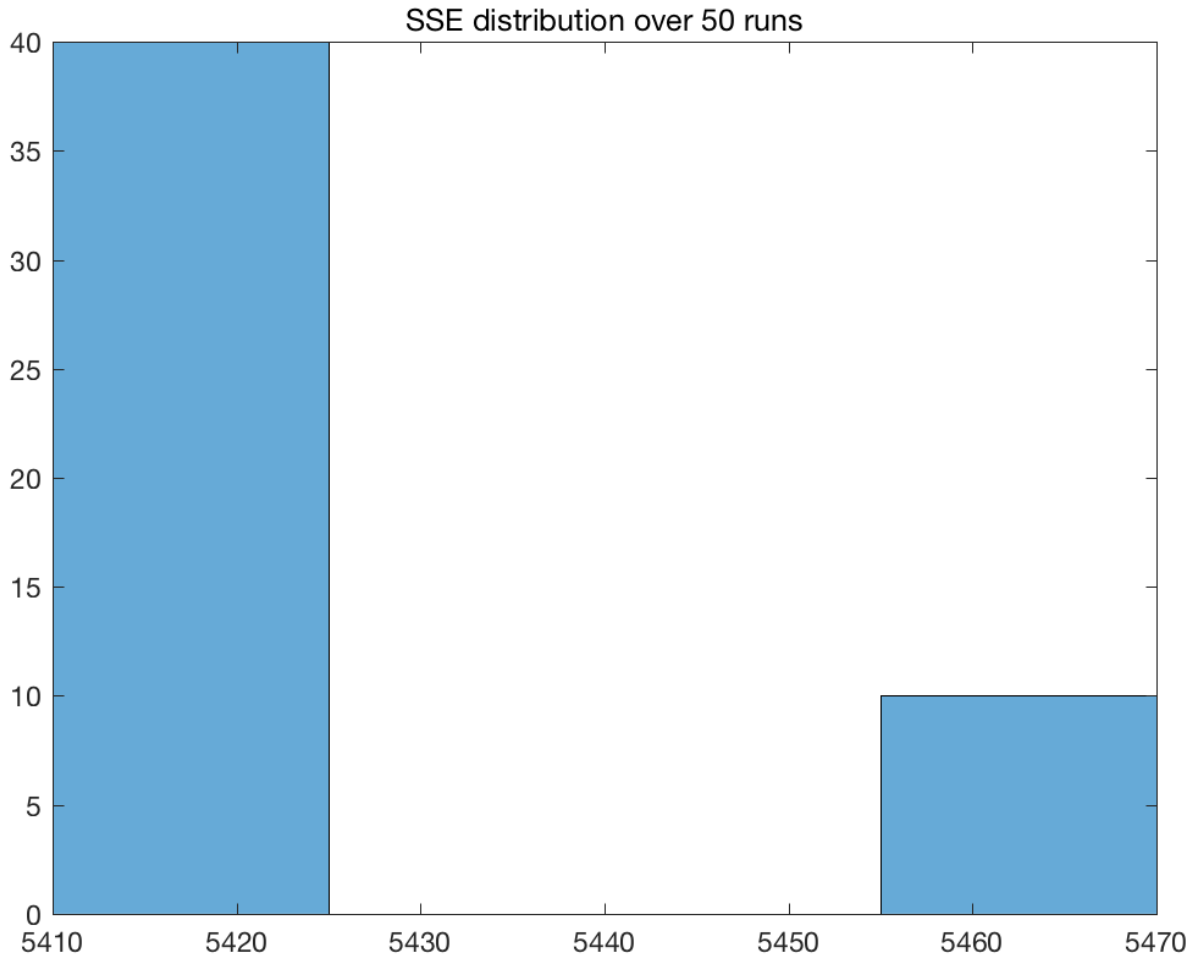


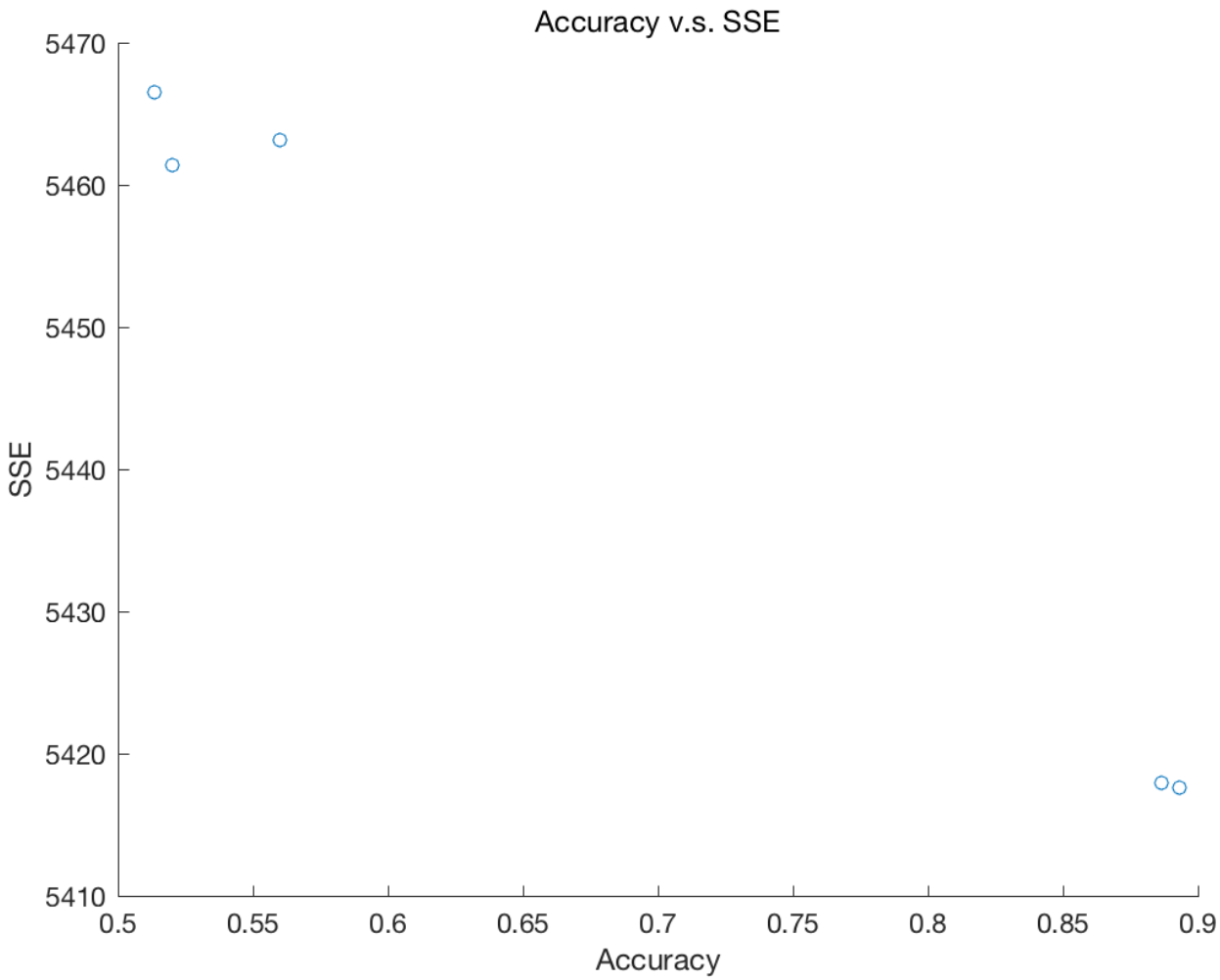
Test clustering quality on the full iris data

- Three accuracy numbers: 0.6667, 0.8867, 0.8867;
- Three RandIndex numbers: 0.8737, 0.8797, 0.8797.

Compare accuracy, RandIndex, and SSE as cluster quality measures


Accuracy distribution over 50 runs
40
35
30
25
20
15
10
5
0
0.45
0.5
0.55
0.6
0.65
0.7
0.75
0.8
0.85
0.9
0.95

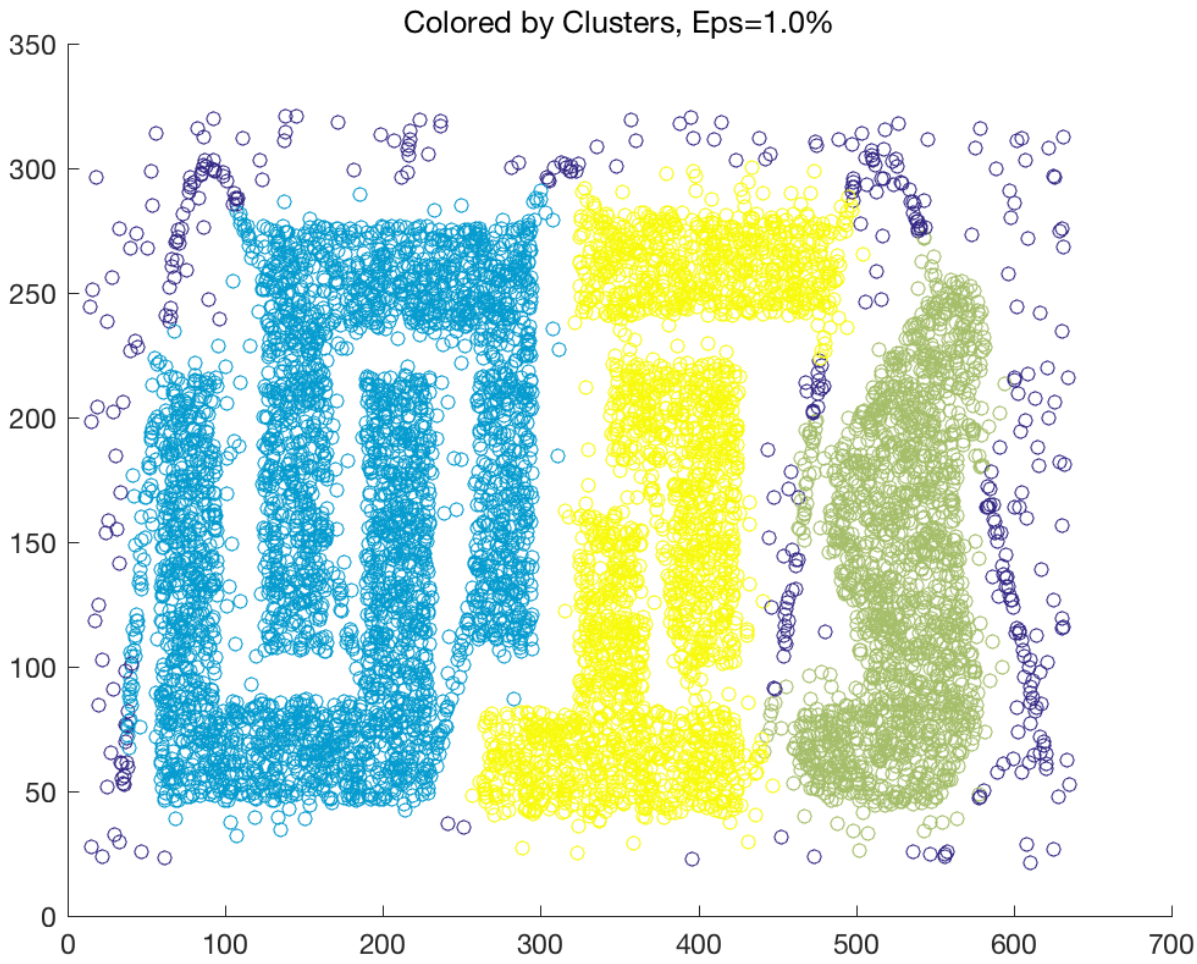

SSE distribution over 50 runs
40
35
30
25
20
15
10
5
0
5410
5420
5430
5440
5450
5460
5470

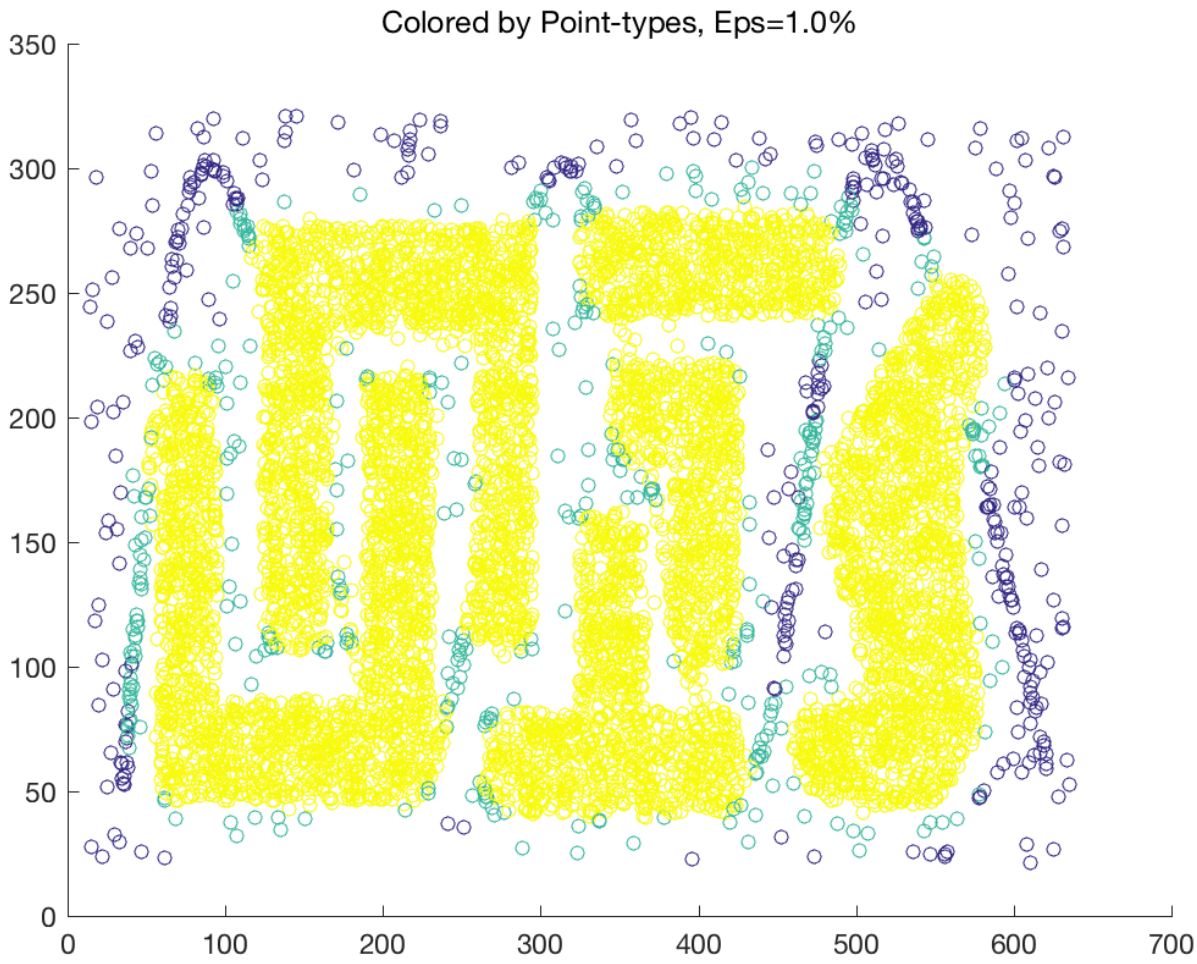


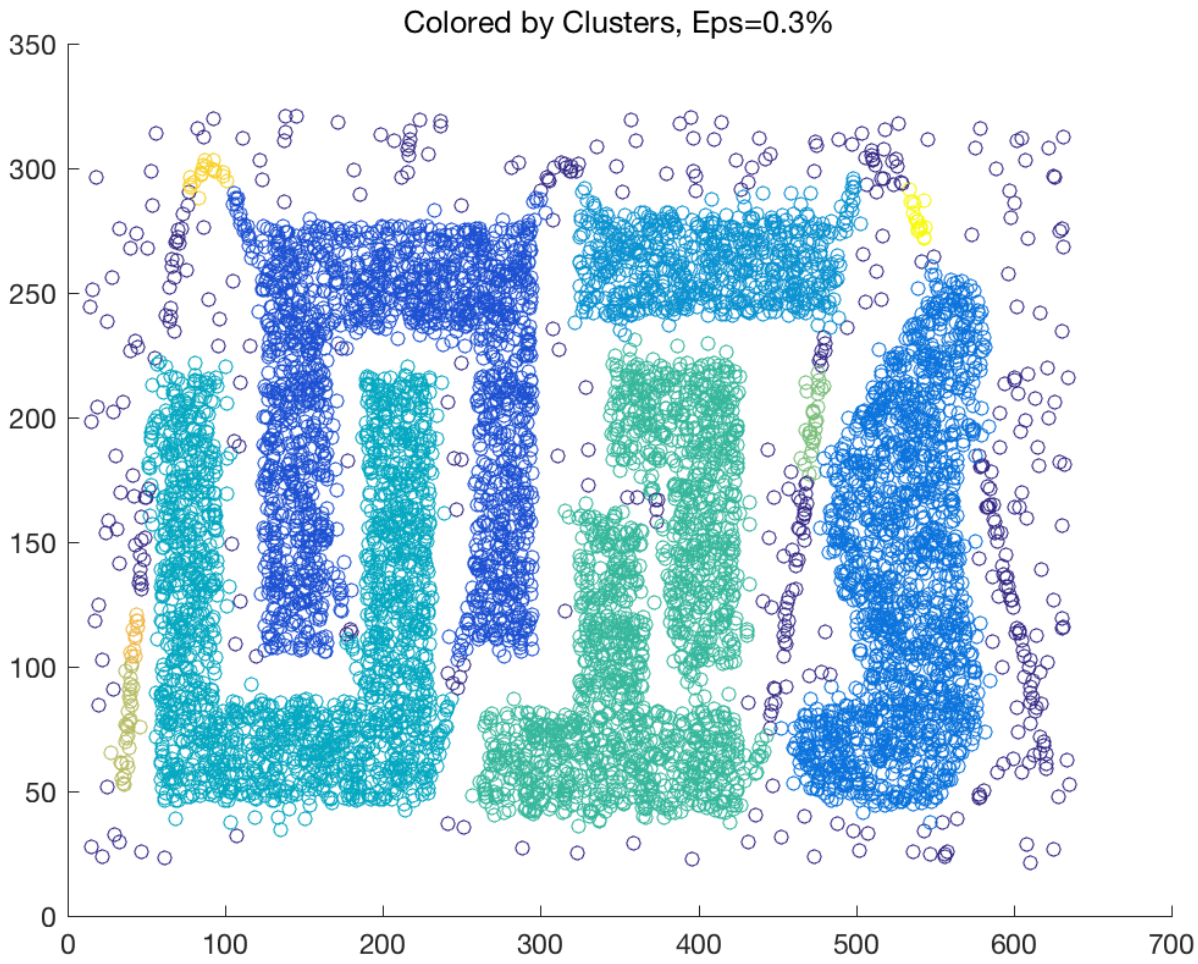
Conclusions

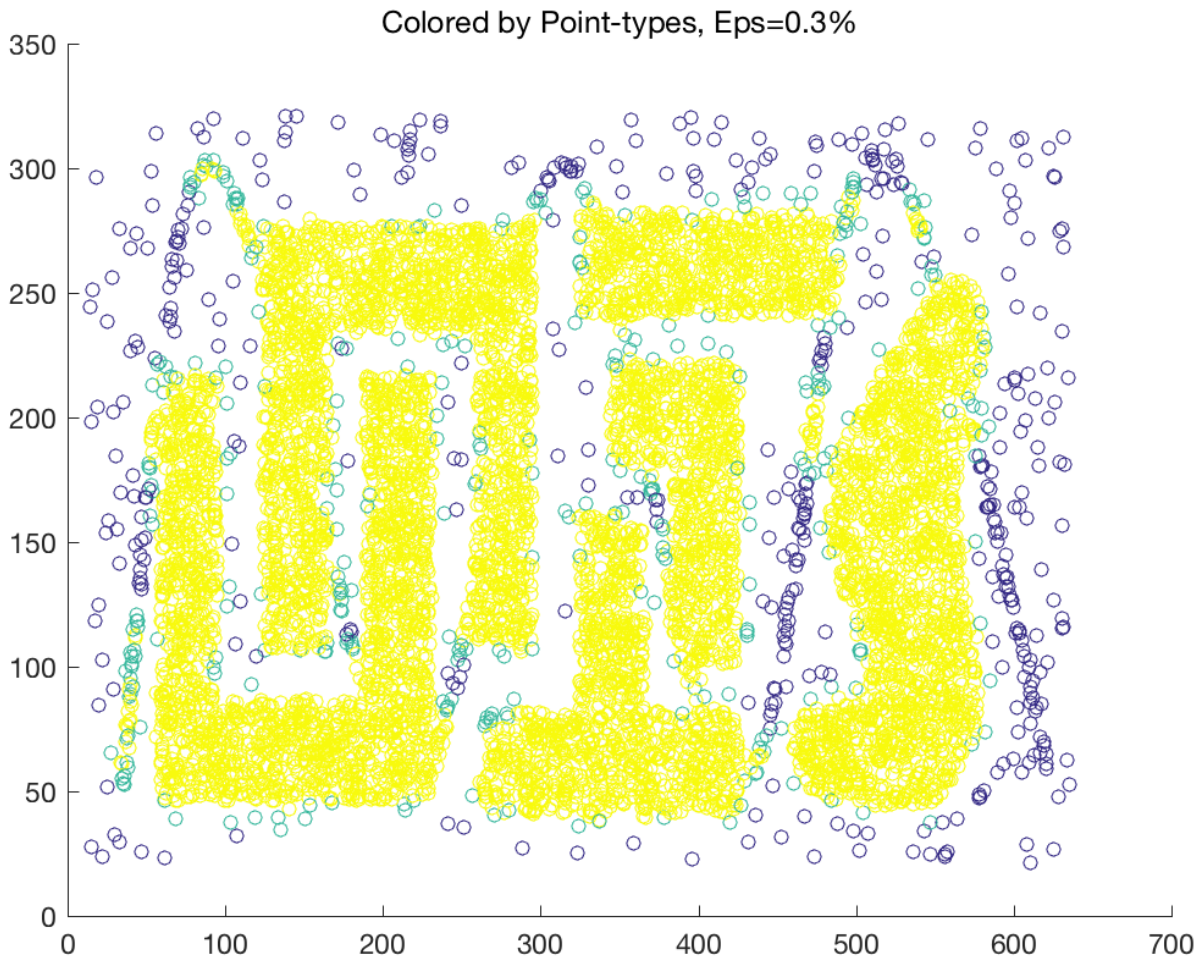
SSE is as good a metric as accuracy in measuring performance of unsupervised learning problems such as clustering. When SSE is low, the points tend to gather close to their centroids, forming a well-separated shape of clusters, leading to high accuracy, and often vice versa. SSE is even better to use when there are no actual labels provided, under what circumstances accuracy cannoe be calculated, but we can still use SSE to obtain the concept of how points congregate (loose or tight, etc.).

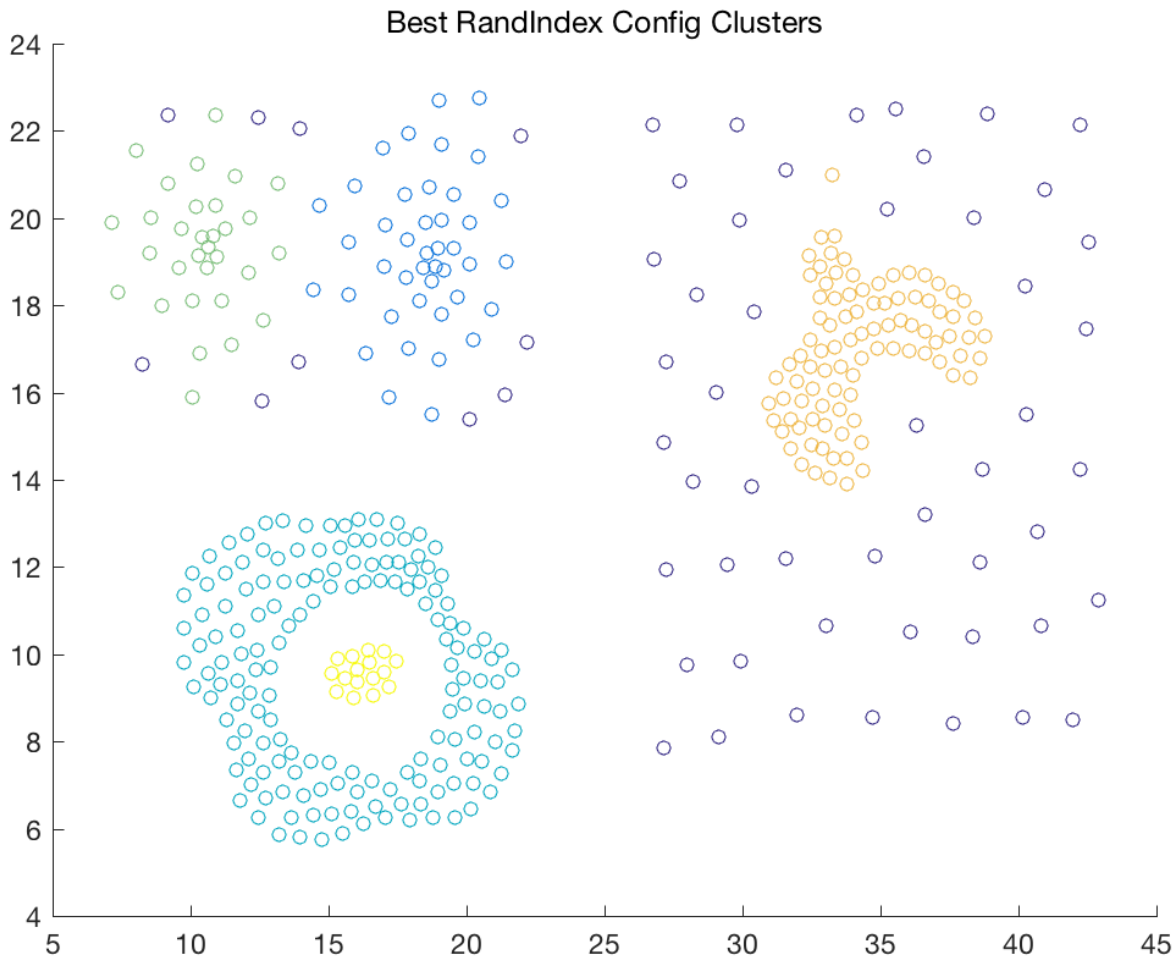
Problem 6

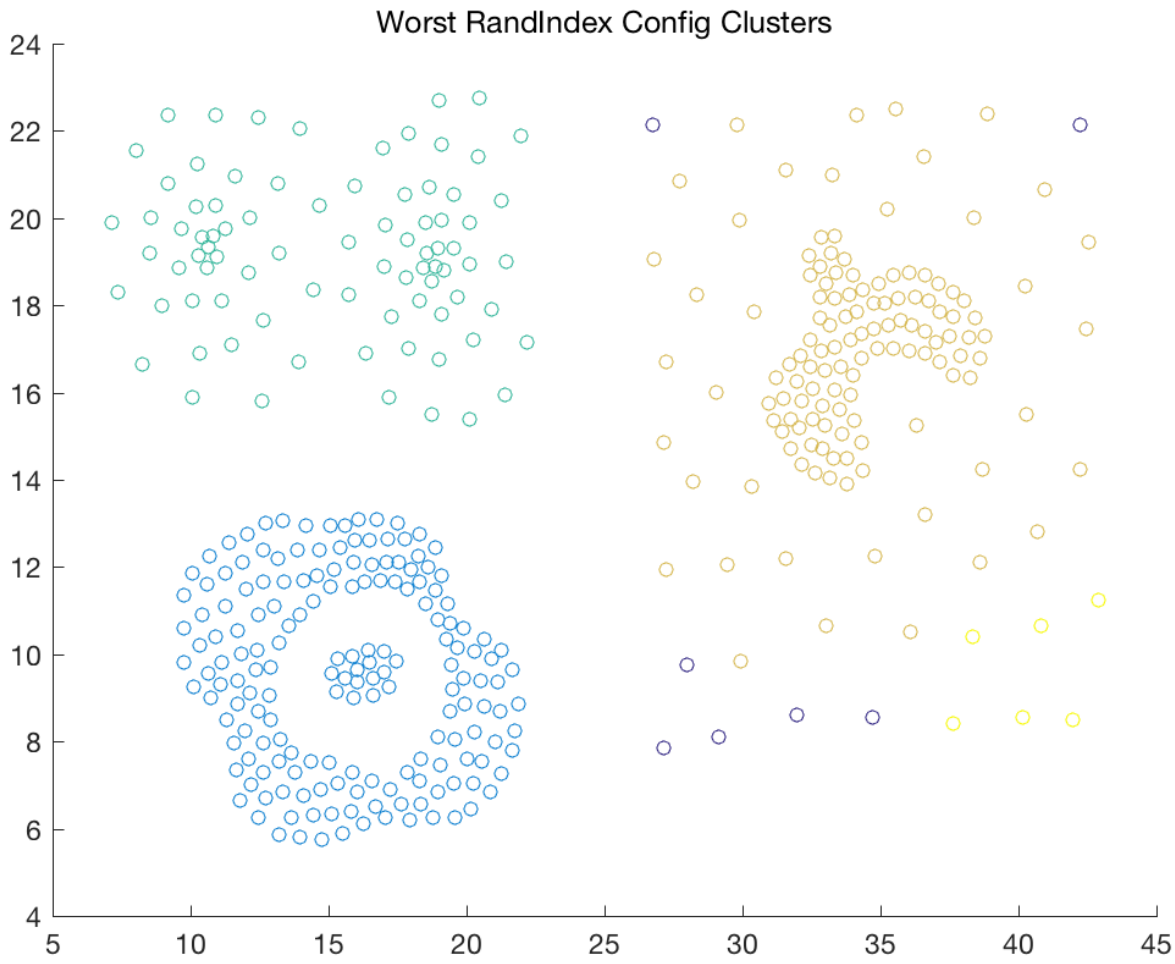

Colored by Clusters, Eps=1.0%
350
300
250
200
150
100
50
0
0
100
200
300
400
500
600
700


Colored by Point-types, Eps=1.0%
350
300
250
200
150
100
50
0
0
100
200
300
400
500
600
700


Colored by Clusters, Eps=0.3%
350
300
250
200
150
100
50
0
0
100
200
300
400
500
600
700


Colored by Point-types, Eps=0.3%
350
300
250
200
150
100
50
0
0
100
200
300
400
500
600
700


Best RandIndex Config Clusters
24
22
20
18
16
14
12
10
8
6
4
5
10
15
20
25
30
35
40
45


Worst RandIndex Config Clusters
24
22
20
18
16
14
12
10
8
6
4
5
10
15
20
25
30
35
40
45