1) a)

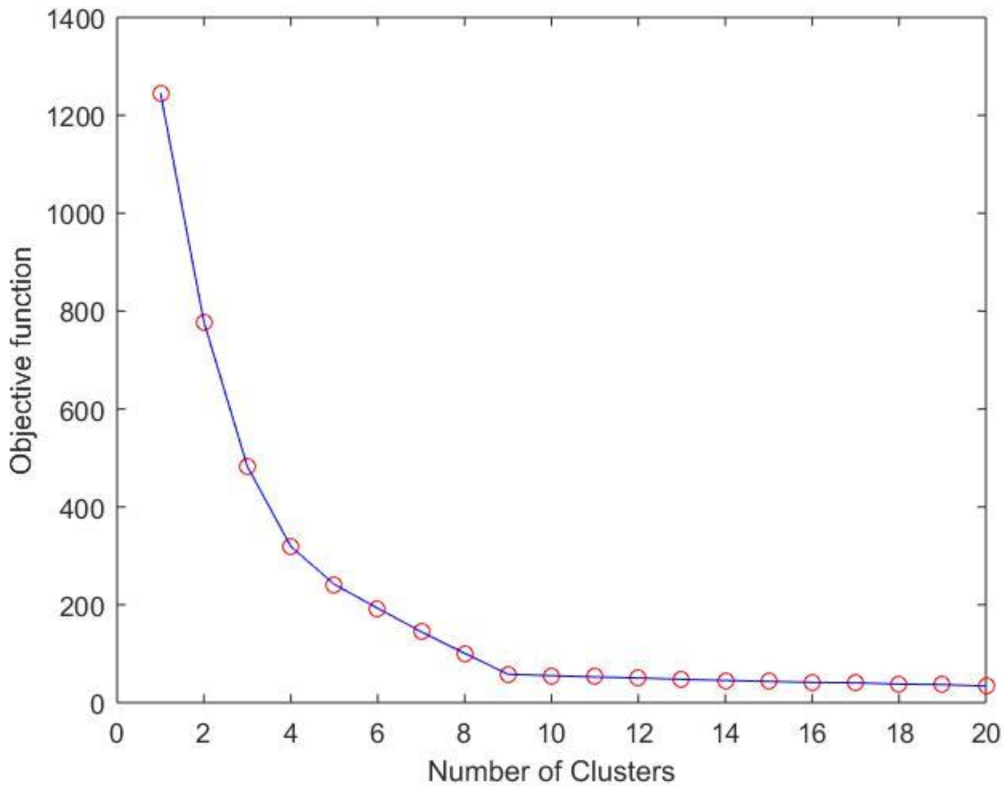
Brute-force algorithm is exponential in "n"

In brute-force, every data point is taken into account to calculate distance from both the cluster centers

$(c_{1i} - x_i)$ and $(c_{2j} - x_j)$
$1 \le i \le n, \quad 1 \le j \le n$

The distance calculation is of the order of n for each cluster center selected (c_{1i}, c_{2j}). But the cluster centers can be selected c_{1i} and c_{1j} for different values of i and j. Totally, the time turns out be order of n^2

1 f)




```
clear
load('kmeans_data.mat')
for i=1:20
    [~,~,obj(i)]=kmeans_cluster(X,i,'random',20) ;
end

[~,~,obj(1)]=kmeans(X,1) ;
plot(obj,'ro'); hold on
plot(obj,'-b') ; xlabel('Number of Clusters') ; ylabel('Objective function')
```

1 g)

Based the above plot, the elbow appears at 9. So, k=9 is the best value for the number of clusters.

This is in agreement with the data, since there are 9 clusters in the data.

Experiment 2: The effect of initialization on Lloyd's method

1 h) The average value of objective function for random is 87.98, when run over N=1000 runs for k=9.

The average value of objective function for kmeans++ is 77.3418, when run over N=1000 runs for k=9.

```
for i=1:1000
    [~,~,obj2(i)]=kmeans_cluster(X,9,'random',1) ;
end
oj = mean(obj2) ;
```

1 i) K-means ++ takes centers based on the probability of, distance of points from current nearest centers. So, higher the distance, more is the probability of the point will be selected as the center.

So the points will be selected, which are farther from current cluster center. This shows that, it is more likely that, one point will be selected from each cluster, since this ensures maximum distance from each centers. The presence of centers in the same cluster, gives rise to minimum distance between each other.

Problem 2 :-

a) True.

For a realizable problem, version space contains all the hypothesis consistent with the labels so far. So, it will atleast contain h^*. So, $H_t \neq \phi$.

b) False.

Considering the example of a real line threshold, It is possible that ~~observing~~ few seen lables will give the hypothesis

$$\begin{array}{ccccccccccc} & & & - & - & - & + & & & \\ -n & -n+1 & \cdots & -1 & 0 & 1 & 2 & \cdots & n-1 & n. \end{array}$$

$$|H| = 2n+1$$

If for example, 2 label is seen, the $|H_n| \neq 1$, in this case

c) True.

After quering x_0 from disaggrement space, some hypothesis which labels it incorrectly will be removed from the version space.

Problem 3: Parity function

a) Since it is a n-bit binary string, there will be "n" values in the string.
   Total number of parity function is given by, 2^n.
   VC-dimension of the parity function is given by,

$$VC\dim(H_{parity}) \le \log_2(2^n) \le n$$

b) Let $e_i = < 0...010...0 >$ For any bits assignment $b_1,...,b_n$ for the vectors $e_1,...,e_n$ we choose the set $S = \{i \mid b_i = 1\}$ We get $h_S(e_j) = \begin{cases} 0 & j \in S \\ 1 & j \notin S \end{cases}$

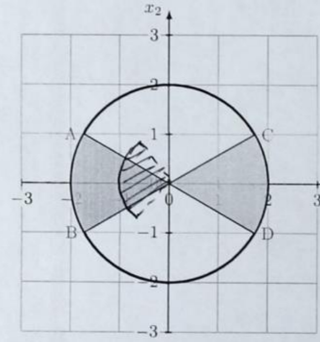
   and so $e_1,...,e_n$ is shattered. So, $VC\dim(H_{parity}) \ge n$

   From the results of a) and b), $VC\dim(H_{parity}) = n$

3)

c) Lets start with one example. In the first example, which is taken in the disagreement space, The string be 00..1011.. ~~This~~ will be removed in this case. Similarly as we go through each label, hypothesis that represent each label incorrectly will be removed. Finally after n labels, the hypothesis that gets all labels correct will be the final.

d) $O\left(\frac{2^n}{\epsilon}\right)$.



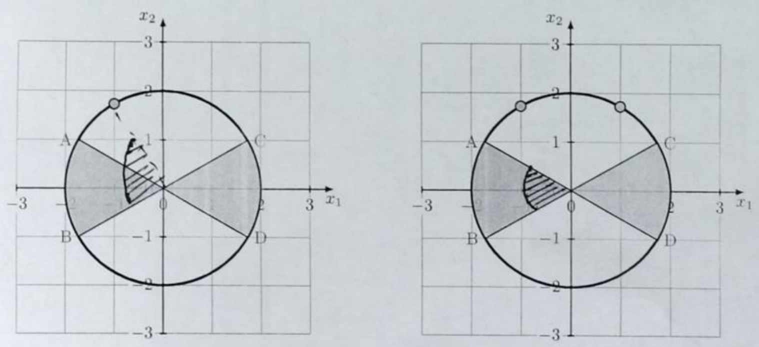
$H_1 \Rightarrow w = \begin{bmatrix} -\sqrt{1-a^2} \\ a \end{bmatrix}, -\frac{\sqrt{3}}{2} < a < \frac{\sqrt{3}}{2}$

Figure 2: All hypotheses in the version space label the blue region as +1 and the red region as -1.

(d) Explicitly write down the version space $H_1 \subseteq H$ that is consistent with all the instances in sectors AOB and COD and draw H_1 in Fig. 2. Specifically, in Fig. 2, shade in the region that corresponds to the set of vectors $\mathbf{w}$ such that $h_{\mathbf{w}} \in H_1$.

(e) Suppose we receive a new instance $(\mathbf{x}, y) = ((-1, \sqrt{3}), +1)$. Write down the new version space $H_2 \subseteq H$ that is consistent with all the instances in both sectors as well as the new instance. Draw H_2 in Fig. 3a.

(f) This time another new instance with negative label $(\mathbf{x}, y) = ((1, \sqrt{3}), -1)$ has been received. Write down the new version space $H_3 \subseteq H$ that is consistent with all the instances in both sectors along with the two new instances. Draw H_3 in Fig. 3b.



(a) A new instance $((-\frac{1}{2}, \frac{\sqrt{3}}{2}), +1)$ is received.

(b) Another new instance $((1, \sqrt{3}), -1)$ is received.

Figure 3

$H_1 \Rightarrow w = \begin{bmatrix} -\sqrt{1-a^2} \\ a \end{bmatrix}, -\frac{1}{2} < a < \frac{\sqrt{3}}{2}$

$H_1 \Rightarrow w = \begin{bmatrix} -\sqrt{1-a^2} \\ a \end{bmatrix}, -\frac{1}{2} < a < \frac{1}{2}$

Problem 4: K-means on the real line

a) The optimal cluster centers will be $k_1 = 1$, $k_2 = 3$, $k_3 = 6.5$. So the objective function will be, 0.5

b) The suboptimal cluster centers can be $k_1 = 2$, $k_2 = 6$, $k_3 = 7$. The objective function value will be 2. The point x_1 and x_2 belong to 1, x_3 belong to 2 and x_4 belong to 3.

   In the next iteration, Current assignment will have the lowest distance for each point. So, the assignment will not be changed for the points.

   Because, the distance of points x_1 from 1^{st} ,2^{nd} and 3^{rd} cluster centers are 1 , 5 and 6 respectively. So, it belongs to cluster 1 only. Its assignment won't change.

   The distance of points x_2 from 1^{st} ,2^{nd} and 3^{rd} cluster centers are 4 , 0 and 1 respectively. So, it belongs to cluster 2 only. Its assignment won't change.

   The distance of points x_3 from 1^{st} ,2^{nd} and 3^{rd} cluster centers are 5 , 1 and 0 respectively. So, it belongs to cluster 3 only. Its assignment won't change.

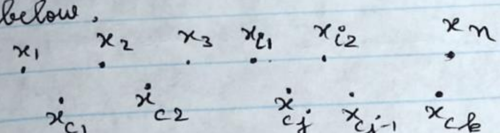
Part 4 :-

c) For each cluster j, there exists i_1 and i_2 such that, cluster consist of $\{x_{i_1}, x_{i_1+1}, \ldots x_{i_2}\}$.

The data points are $x_1 \le x_2 \le \cdots \le x_n$.

Let the cluster center points be represented by, $x_{c1}, x_{c2}, \ldots x_{cj}, x_{cj-1}, \ldots x_{ck}$.
There are k clusters, with $k < n$.

Consider these points in the line as shown below.



Let this be the optimal cluster assignment

for x_{cj}, there will be set of point $x_{i_1}, x_{i_1+1}, \ldots x_{i_2}$ belonging to the j^{th} cluster with $x_{i_1} \le x_{i_1+1} \ldots \le x_{i_2}$. All the points in the range $[x_{i_1}, x_{i_2}]$ belongs to the cluster j, because, if x_{i_1} and x_{i_2} belongs to j,

then $|x_{cj} - x_{i_1}| < |x_{cl} - x_{i_1}|$, for all $l \in [1, \ldots k)$ except j
i.e. its distance from other centers will be higher

Similarly,

$$|x_{c_j} - x_{i_2}| < |x_{c_l} - x_{i_2}|, \text{ for all } l \in [1..k] \text{ except } j$$

If x_{i_1} and x_{i_2}, follows this property, then all points between x_{i_1} and x_{i_2} has to follow the property, because

$$x_{i_1} \le x_{i_1+1} \ldots x_{i_2}$$

So, all the points in the range of $[x_{i_1}, x_{i_2}]$ belongs to cluster j.

d) dynamic programming:-

1-D kmeans, can be defined as assigning the data into k clusters, so that squares of within-cluster distances from each element is minimized.

Now, minimization of within cluster distances can be considered as sub-problem, where we cluster $x_1, \ldots x_i$ into m clusters.

Let $D[i, m]$ be matrix storing min. sum of square of $x_1, \ldots x_i$ into m clusters.

$D[n, k]$ is min. sum of square for the problem

Let, j be index of smallest number in cluster m in optimal solution to $D[i,m]$. This show that $D[j-1, m-1]$ must be the optimal sum of square for first $j-1$ points into $m-1$ clusters. This leads to recurrance equation,

$$D[i,m] = \min_{m \le j \le i} \left\{ D[j-1, m-1] + d(x_j, \dots x_i) \right\}.$$

$d(x_j, \dots x_i) \Rightarrow$ This is the sum of square dist of the points to their mean.

$D[n,m]$ requires $O(n^2)$ time to compute.

$$d(x_1, \dots x_i) = d(x_1, \dots x_{i-1}) + \frac{i-1}{i}(x_i - \mu_{i-1})^2$$

The overall time required is $O(n^2 k)$