Assignment-1

Answer 1.

$P_n(R) = \{$ set of all polynomials in x, with real coefficients $\}$

$\Rightarrow P_n(R) = a_0 + a_1x + a_2x^2 \dots a_nx^n$; $a_1, a_2 \dots a_n \in R$

Claim: $P_n(R)$ is a vector space.

Proof: $\rightarrow$ Addition : $p = a_0 + a_1x \dots a_nx^n$

$q = b_0 + b_1x \dots b_nx^n$

$\Rightarrow p+q = (a_0+b_0) + (a_1+b_1)x \dots (a_n+b_n)x^n$

* $\Rightarrow p+q \in P_n(R)$

* $(a_0 + a_1x \dots a_nx^n) + (-a_0 + (-a_1)x \dots (-a_n)x^n) = 0$

* $(a_0 + a_1x \dots) + b_0 + b_1x \dots) + (c_0 + c_1x \dots)$

$= (a_0 + a_1x \dots) + (b_0 + b_1x \dots + c_0 + c_1x \dots)$

$= (a_0 + b_0 + c_0)x + \dots$

* $(a_0 + a_1x \dots a_nx^n) + 0 = a_0 + a_1x \dots$

* $(a_0 + b_0) + (a_1 + b_1)x \dots = (b_0 + a_0) + (b_1 + a_1)x \dots$

$\rightarrow$ Scalar Multiplication

* $\alpha(a_0 + a_1x \dots) = \alpha a_0 + (\alpha a_1)x \dots \in P_n(R)$

* $(\alpha\beta)(a_0 + a_1x \dots) = (\alpha\beta a_0) + (\alpha\beta a_1)x \dots$

$= \alpha[\beta a_0 + (\beta a_1)x \dots]$

* $1 * (a_0 + a_1x \dots) = a_0 + a_1x \dots$

* $(\alpha+\beta)(a_0 + a_1x \dots) = \alpha(a_0 + a_1x \dots) + \beta(a_0 + a_1x \dots)$

* $\alpha(a_0 + a_1x \dots + b_0 + b_1x \dots) = (\alpha a_0 + \alpha a_1x \dots) + (\alpha b_0 + \alpha b_1x \dots)$

$\Rightarrow P_n(R)$ is a vector space.

b.

$$F(p(x)) = \frac{d}{dx} p(x) \Big|_{x=0}$$

$$F(\alpha p(x) + \beta q(x)) = \frac{d}{dx}(\alpha p(x) + \beta q(x)) \Big|_{x=0}$$

$$= \alpha \frac{d}{dx} p(x) \Big|_{x=0} + \beta \frac{d}{dx} q(x) \Big|_{x=0}$$

$$= \alpha F(p(x)) + \beta F(q(x))$$

$\Rightarrow F(p(x))$ is a linear functional.

Proved.

c. $p(x) = a_0 + a_1 x \cdots a_n x^n$

$$\Rightarrow p = \begin{bmatrix} a_0 \\ \vdots \\ \vdots \\ a_n \end{bmatrix}$$

$$F[p(x)] = a_1 + 2a_2 x \cdots n a_n x^{n-1} \Big|_{x=0}$$

$$= a_1$$

$$\therefore F[p(x)] = \boxed{e_1^T p \quad \text{Ans.}}$$

Answer 2.

$$\operatorname{avg}(x) = \left(\frac{1^n}{n}\right)^T x \qquad \operatorname{std}(x) = \frac{\|x - \operatorname{avg}(x) 1_n\|_2}{\sqrt{n}}$$

$$= \frac{1}{n}(1^n)^T x$$

a. $$\operatorname{avg}(\alpha x + \beta 1_n) = \left(\frac{1^n}{n}\right)^T [\alpha x + \beta 1_n]$$

$$= \left(\frac{1^n}{n}\right)^T \alpha x + \left(\frac{1^n}{n}\right)^T \beta 1_n$$

$$= \alpha \underbrace{\left(\frac{1^n}{n}\right)^T x}_{\operatorname{avg}(x)} + \frac{\beta}{n} \underbrace{(1^n)^T 1^n}_{n}$$

$$= \alpha \operatorname{avg}(x) + \beta$$ Proved.

b. $$\operatorname{std}(\alpha x + \beta 1_n) = \frac{\|\alpha x + \beta 1_n - \operatorname{avg}(\alpha x + \beta 1_n) 1_n\|_2}{\sqrt{n}}$$

$$= \frac{\|\alpha x + \beta 1_n - \alpha \operatorname{avg}(x) 1_n - \beta 1_n\|_2}{\sqrt{n}}$$

$$= \frac{\|\alpha x - \alpha \operatorname{avg}(x) 1_n\|_2}{\sqrt{n}}$$

$$= |\alpha| \frac{\|x - \operatorname{avg}(x)\|_2}{\sqrt{n}} = |\alpha| \operatorname{std}(x)$$ Proved.

Answer 3.

3. $$||x||_\omega = \sqrt{\sum_{i=1}^{n} w_i x_i^2}$$

* Homogeinty

$$||x||_\omega = \sqrt{\omega_1 x_1^2 + \omega_2 x_2^2 \dots w_n x_n^2}$$

$w_i \in R$, $x_i \in R$ $\forall i$

$$\Rightarrow ||x||_\omega \in R$$

$$||.|| : R^n \rightarrow R$$

* Non- negativity

$$||x||_\omega = \sqrt{\omega_1 x_1^2 + \omega_2 x_2^2 \dots \omega_n x_n^2}$$

$$w_i > 0 \;, \; x_i^2 \geqslant 0$$

$$\Rightarrow ||x||_\omega \geqslant 0$$

* Definiteness

$$||x||_\omega = \sqrt{\omega_1 x_1^2 + \omega_2 x_2^2 \dots \omega_n x_n^2} = 0$$

$$\Rightarrow w_1 x_1^2 = w_2 x_2^2 = \dots = w_n x_n^2 = 0$$

$$w_1, w_2 \dots w_n > 0$$

$$\Rightarrow x_1 = x_2 = x_3 \dots = 0$$

$$\Rightarrow x = 0$$

* Triangle Inequality

$$f : R^n \rightarrow R^n \Rightarrow \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \rightarrow \begin{bmatrix} \sqrt{w_1}\, x_1 \\ \vdots \\ \sqrt{w_n}\, x_n \end{bmatrix}$$

$$x \rightarrow y$$

from Δ-inequality of l_2 norm

$$\therefore ||x||_\omega = ||y||_2 \Rightarrow ||y_1 + y_2||_2 \leqslant ||y_1||_2 + ||y_2||_2$$

$$\Rightarrow ||(x_1 + x_2)||_2 \leqslant ||x_1||_\omega + ||x_2||_\omega$$

$$y_1+y_2 = \begin{bmatrix} \sqrt{w_1}\,(x_1+x_1') \\ \sqrt{w_2}\,(x_2+x_2') \\ \vdots \\ \vdots \end{bmatrix} \Rightarrow \|y_1+y_2\|_2 = \|(x+x')\|_w$$

$\Rightarrow \|\cdot\|_w$ is a norm called weighted norm.

Proved.

Answer 4.

a.

Step 1: for all $x_1 \ldots x_N$

$\rightarrow$ find $\|x_1-z_1\|_2 \ldots \|x_1-z_k\|_2$: subtraction $\approx n$, norm $\approx 2n$ — $(3n)^* k$

$\rightarrow$ find min among them : k — $(3n+1)^* k$

$\Rightarrow$ Total complexity : $(3n+1)^* k^* N$

$\approx \boxed{(nkN) \text{ Ans.}}$

b.

Step 2: for all $j = 1 \ldots k$,

$\rightarrow \quad z_j = \frac{1}{|C_j|} \sum x_i \{\text{s.t. } c_i = j\}$

Total sum $= n^* N$

no. of divisions $= k$

Total complexity $= \boxed{n^* N + k \text{ Ans.}}$

Combining Step 1 and Step 2,

$$\text{complexity} \approx nkN + nN + k$$

(as $k < n$, generally)

$$\approx nN$$

$\therefore$ No. of computation for iters = 10

$\Rightarrow \boxed{10nN \text{ Ans.}}$

Answer 5.

Link to code: https://gist.github.com/sriyash421/9c2b05e9ba1a80e1d7ccdab3035f4955

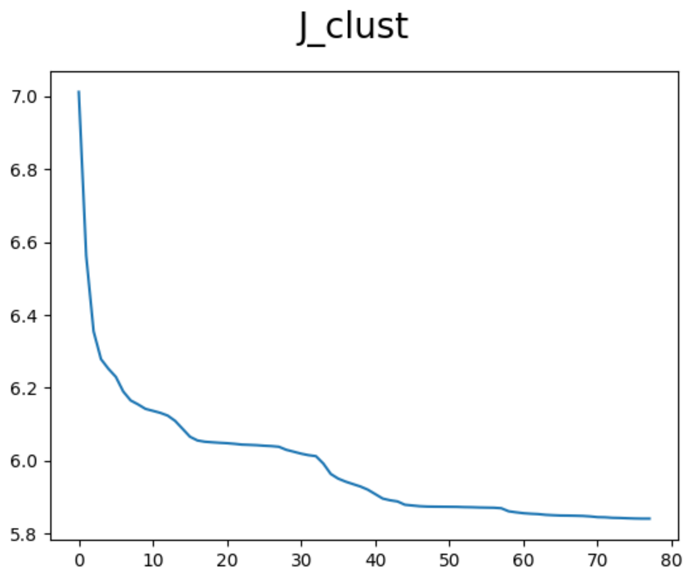
Training samples :

```
x_train: (1000, 784) y_train: (1000,) x_test: (50, 784) y_test: (50,)
No. of training samples, N: 1000
Length of vector, n: 784
```

Convergence criterion: If the value of J_clust doesn't change over an iteration.

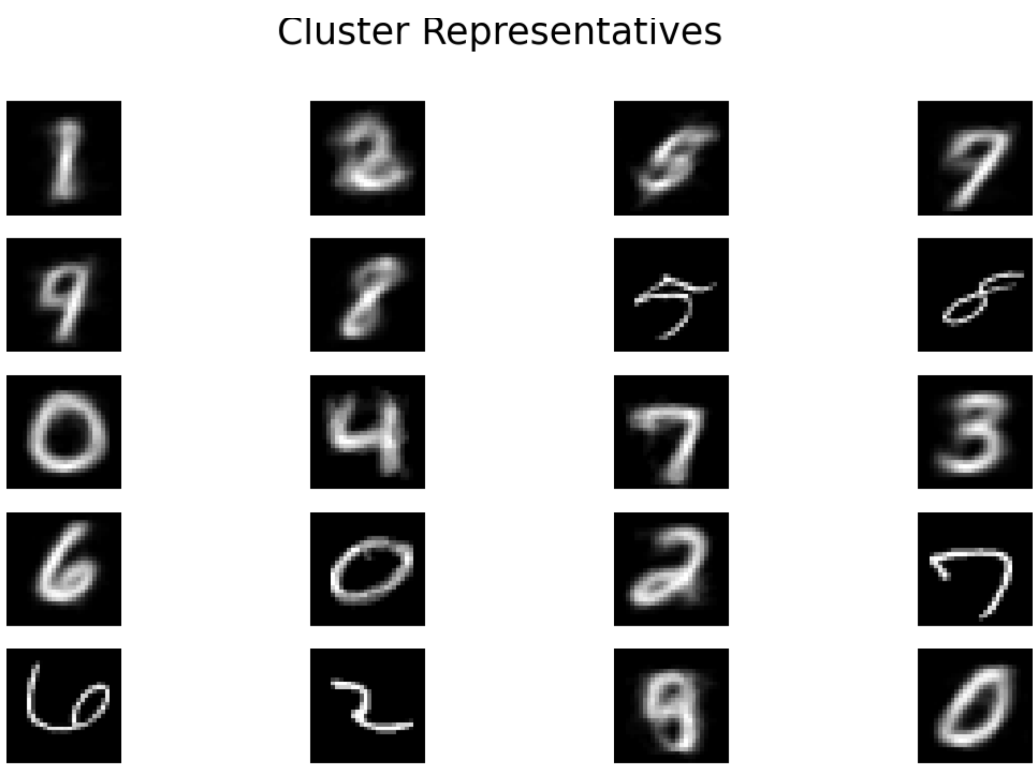
```
def convergence_criterion(self):
    if len(self.loss) < 2:
        return False
    if self.loss[-1] == self.loss[-2]:
        return True
    return False
```

(i). Random initialisation

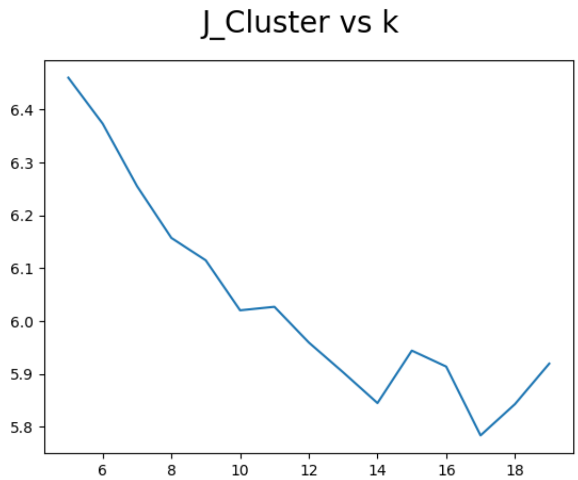


The plot of loss vs iteration

```
Converged at iteration: 78
J_clust: 5.841378379925047
Accuracy: 0.5199999809265137
```



Cluster representatives after convergence

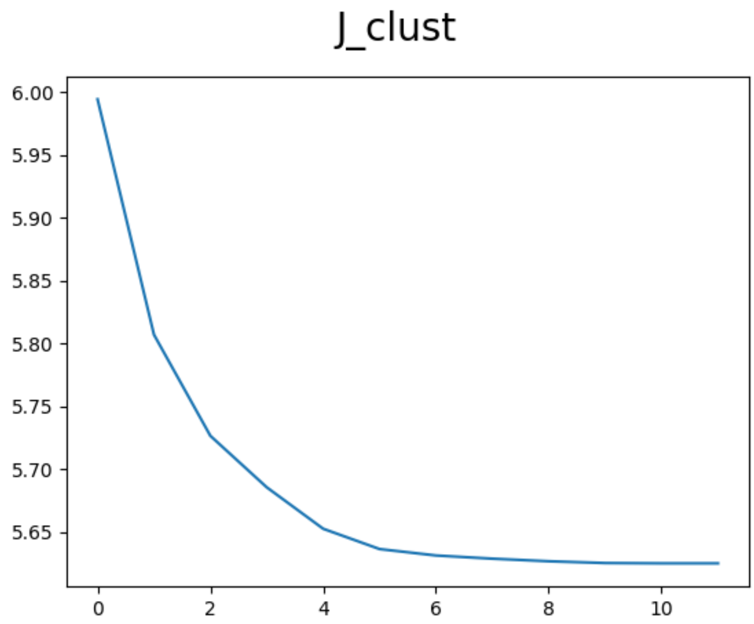


A plot of J_clust on convergence vs k

```
Min J_clust value: 5.784021688919853 at k= 17
```

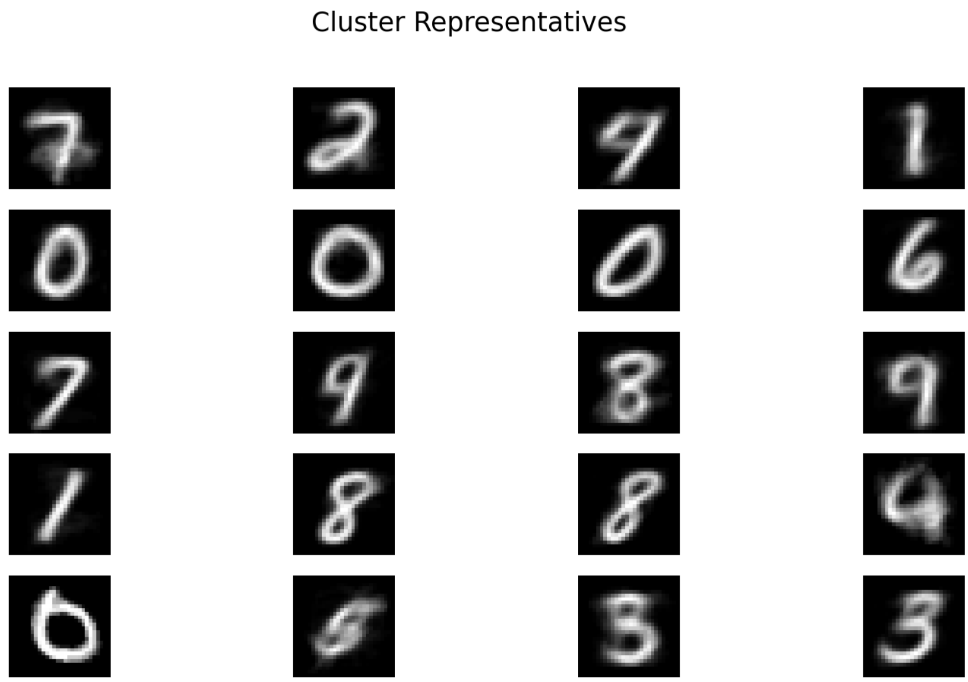
Here, the optimal K-value is 17, as the value of J_clust is a min at that value. The optimal number of clusters is more than 10, in spite of the number of classes being exactly 10, which is mainly due to the different ways of writing different digits.
Also cluster representatives in initialization from training samples is closer to actual nums as compared to the other method.

(ii). Initialization from training values

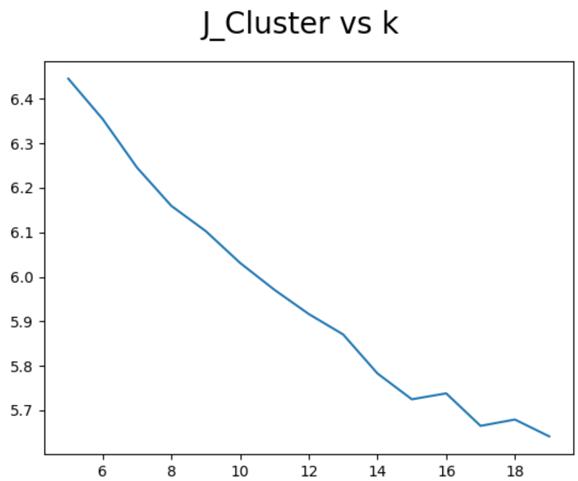


The plot of loss vs iteration

```
Converged at iteration: 12
J_clust: 5.624799552168552
Accuracy: 0.5799999833106995
```



Cluster representatives after convergence



A plot of J_clust on convergence vs k

```
Min J_clust value: 5.64137564798474 at k= 19
```

Here, the optimal K-value is 19, as the value of J_clust is a min at that value. The optimal number of clusters is more than 10, in spite of the number of classes being exactly 10, which is mainly due to the different ways of writing different digits.

Yes, the initial condition choice affects the number of iterations of convergence, the final accuracy, and the final J-cluster value. Accuracy is slightly higher, J_clust is slightly lower, and num_iterations is much lower when we initialize the cluster representations from the training values. So, given the empirical evidence, it is optimal to choose the initial representatives from the training samples.