Subject:
Date

یادگیری ماشین تمرین پنجم مهدی جمال خواه
810100111

سوال ① :
الف)

$$X_1 = \begin{bmatrix} 4 & 2 & 2 & 3 & 4 \\ 1 & 4 & 3 & 6 & 4 \end{bmatrix} \Rightarrow m_1 = \begin{bmatrix} 3 \\ 3.6 \end{bmatrix}$$

$$X_2 = \begin{bmatrix} 9 & 6 & 9 & 8 & 10 \\ 10 & 8 & 5 & 7 & 8 \end{bmatrix} \Rightarrow m_2 = \begin{bmatrix} 8.4 \\ 7.6 \end{bmatrix}$$

$$(X_1 - m_1) = \begin{bmatrix} 1 & -1 & -1 & 0 & 1 \\ -2.6 & 0.4 & -0.6 & 2.4 & 0.4 \end{bmatrix}$$

$$(X_2 - m_2) = \begin{bmatrix} 0.6 & -2.4 & 0.6 & -0.4 & 1.6 \\ 2.4 & 0.4 & -2.6 & -0.6 & 0.4 \end{bmatrix}$$

$$S_1^2 = \begin{bmatrix} 1 & -1 & -1 & 0 & 1 \\ -2.6 & 0.4 & -0.6 & 2.4 & 0.4 \end{bmatrix} \cdot \begin{bmatrix} 1 & -2.6 \\ -1 & 0.4 \\ -1 & -0.6 \\ 0 & 2.4 \\ 1 & 0.4 \end{bmatrix} = \begin{bmatrix} 4 & -2 \\ -2 & 13.2 \end{bmatrix}$$

$$S_2^2 = \begin{bmatrix} 0.6 & -2.4 & 0.6 & -0.4 & 1.6 \\ 2.4 & 0.4 & -2.6 & -0.6 & 0.4 \end{bmatrix} \begin{bmatrix} 0.6 & 2.4 \\ -2.4 & 0.4 \\ 0.6 & -2.6 \\ -0.4 & -0.6 \\ 1.6 & 0.4 \end{bmatrix} = \begin{bmatrix} 9.2 & -0.2 \\ -0.2 & 13.2 \end{bmatrix}$$

PAPCO

Subject:
Date

$$S_w = S_1^2 + S_2^2 = \boxed{\begin{bmatrix} 13.2 & -2.2 \\ -2.2 & 26.4 \end{bmatrix}} \blacksquare$$

ب)

$$m_1 - m_2 = \begin{bmatrix} -5.4 \\ -4 \end{bmatrix}$$

$$S_B = (m_1 - m_2)(m_1 - m_2)^T = \begin{bmatrix} -5.4 \\ -4 \end{bmatrix} \begin{bmatrix} -5.4 & -4 \end{bmatrix}$$

$$= \boxed{\begin{bmatrix} 29.16 & 21.6 \\ 21.6 & 16 \end{bmatrix}} \blacksquare$$

ج)

$$S_w^{-1} = \frac{1}{343.64} \begin{bmatrix} 26.4 & 2.2 \\ 2.2 & 13.2 \end{bmatrix}$$

$$A = S_w^{-1} S_B = \frac{1}{343.64} \begin{bmatrix} 817.344 & 605.44 \\ 349.272 & 258.72 \end{bmatrix} = \begin{bmatrix} 2.37 & 1.76 \\ 1.01 & 0.75 \end{bmatrix}$$

$$|A - \lambda I| = 0 \Rightarrow (2.37 - \lambda)(0.75 - \lambda) = 1.01 * 1.76$$

$$\Rightarrow \boxed{\lambda_1 = 3.13} \blacksquare \qquad \lambda_2 = 6.6 \times 10^{-16} \simeq 0$$

Subject:
Date

سوال ② :

الف) وقتی با یک مساله ML روبرو هستیم با توجه به محدودیت هایمان از جمله زمان، حافظه، سخت افزار کاندید هایی برای انتخاب مدل داریم مثل SVM، MLP و...

حال با توجه به دانش دامنه و مساله و بررسی نتایج هر کاندید یک مدل را انتخاب می کنیم به این کار model selection می گوییم و معمولاً بخشی از داده را به عنوان validation جدا می کنیم و از آن برای ارزیابی مدل های مختلف استفاده می کنیم تا بهترین مدل را انتخاب کنیم. مدل داده ی validation را در زمان آموزش نباید ببیند.

حال که مدلمان را انتخاب کرده ایم باید خطای پیش بینی آن را روی یک داده جدید و مستقل تخمین بزنیم (داده تست که جدای از داده آموزش و validation باید باشد) تا متوجه شویم که مدل چقدر توان Generalization دارد یعنی می تواند داده هایی که تا الان ندیده است را به درستی پیش بینی کند. به این عمل model assessment می گویند.

دلیل به کارگیری از model selection : هر مدل قدرت متفاوت و تعداد پارامتر های متفاوتی دارد، مثلاً بعضی نیاز به داده زیاد برای آموزش دارند که ما با توجه

PAPCO

Subject:
Date

به مساله و داده‌هایمان باید مدل مناسب را پیدا کنیم. نکته مهم‌تر inductive bias

هر مدل است مثلاً بعضی مدل‌ها خطی هستند یعنی هرچقدر هم به آن داده بدهیم

درنهایت کلاس‌ها را با یک خط جدا می‌کند یا بعضی فرض می‌کنند که داده‌ها از یک

توزیع خاص پیروی می‌کنند اگر inductive bias با مساله و دیتاست ما

همخوانی نداشته باشد نتیجه خوبی نخواهیم گرفت و از آن طرف اگر داشته باشد

باعث می‌شود با هزینه‌ی کمتری به همان نتیجه مطلوب برسیم که یک مدل بدون

آن inductive bias (یعنی قدرت بیش‌تری نسبت به مدل قبل دارد) می‌رسد

در نتیجه باید مدل مناسب را پیدا کنیم.

ب) در دید Probabilistic برای امتیاز دهی به مدل کاندید از مدل performance

روی داده آموزش و میزان Complexity استفاده می‌کنیم یعنی اگر performance

روی داده آموزش افزایش پیدا کند امتیاز آن نیز افزایش پیدا می‌کند اما اگر

Complexity مدل افزایش پیدا کند امتیاز آن کاهش پیدا می‌کند.

در دید Resampling سادگی مدل را براساس داده‌ها جدید که مدل تا اکنون ندیده

PAPCO

Subject:
Date

است را می سنجیم.

مقایسه دو روش :

* روش probabilistic روی داده های آموزش پایاس می شود یعنی ممکن است generalization آن کم شود هرچند آن ترم complexity (ترم منظم ساز) که به آن اضافه می شود سعی می کند جلوی آن را بگیرد ، اما همچنان مشکل وجود دارد درحالی که روش Resampling روی داده های جدید آزمایش می شود و این مشکل را ندارد

* از طرفی در روش Resampling باید بخشی از داده هایمان را به عنوان validation کنار بگذاریم و در زمان آموزش از آن استفاده نکنیم و اگر داده به اندازه کافی نداشته باشیم مشکل ساز می شود اما روش probabilistic این مشکل را ندارد.

ج) وقتی داده کم باشد وقتی بخشی را برای validation کناری گذاریم دیگر داده کافی برای آموزش نخواهیم داشت و نمی توان مقدار optimal پارامتر ها را بدست آورد.

چند راه برای حل مشکل می توانیم ارائه دهیم ابتدا از روش های ساده تر و در دسترس

PAPCO

Subject:
Date

شروع می کنیم: می توانیم از cross-validation K-fold استفاده کنیم. به این صورت که داده ها را به K دسته تقسیم می کنیم و هر بار یک دسته را به عنوان validation و K-1 دسته دیگر را به عنوان داده آموزش در نظر می گیریم و در نهایت نتایج این K حالت را میانگین می گیریم. یک راه دیگر استفاده از data augmentation است. یعنی به صورت مصنوعی داده تولید می کنیم. همچنین می توانیم از روش های transfer learning نیز استفاده کنیم یعنی از یک مدل pre train شده استفاده کنیم و با توجه به شکل خودمان آن را fine tune کنیم. روش های نیاز به یک مدل pre train شده است که ممکن است همیشه در دسترس نباشد

اما روش cross-validation K-fold راحت تر و تقریباً همیشه قابل انجام است

اما قطعاً نتیجه روش fine tune را هم نخواهد داشت

PAPCO

Subject:
Date

سوال ③:

$$P(x) = \alpha \lambda_1 e^{-\lambda_1 x} + (1-\alpha)\lambda_2 e^{-\lambda_2 x} \qquad \theta = (\alpha, \lambda_1, \lambda_2)$$

$$P(x, z \mid \theta) = \left[\alpha \lambda_1 e^{-\lambda_1 x}\right]^{z} \left[(1-\alpha)\lambda_2 e^{-\lambda_2 x}\right]^{(1-z)}$$

$$\ell(\theta) = \sum_{i=1}^{n} \log P(x_i, z_i \mid \theta)$$

$$Q(\theta) = E_{z|x}\left[\ell(\theta)\right] = \sum_{i=1}^{n} E_{z|x}\left[\log P(x_i, z_i \mid \theta)\right]$$

$$= \sum_{i=1}^{n} E_{z|x}\left[z_i(\log\alpha + \log\lambda_1 - \lambda_1 x_i) + (1-z_i)\left(\log(1-\alpha) + \log\lambda_2 - \lambda_2 x_i\right)\right]$$

$$= \sum_{i=1}^{n}\left[\underset{z|x}{E}[z_i]\left(\log\alpha + \log\lambda_1 - \lambda_1 x_i\right) + \left(1 - \underset{z|x}{E}[z_i]\right)\left(\log(1-\alpha) + \log\lambda_2 - \lambda_2 x_i\right)\right]$$

$$\underset{z|x}{E}[z] = E[z \mid x] = P(z=1 \mid x) = \frac{P(x \mid z=1)\,P(z=1)}{P(x)}$$

$$= \boxed{\frac{\alpha^t \lambda_1^t e^{-\lambda_1^t x}}{\alpha^t \lambda_1^t e^{-\lambda_1^t x} + (1-\alpha^t)\lambda_2^t e^{-\lambda_2^t x}} = \gamma^t}$$

$$Q(\theta) = \sum_{i=1}^{n}\left[\gamma_i^t\left(\log\alpha + \log\lambda_1 - \lambda_1 x_i\right) + (1-\gamma_i^t)\left(\log(1-\alpha) + \log\lambda_2 - \lambda_2 x_i\right)\right]$$

PAPCO

Subject:
Date

$$\frac{\partial Q}{\partial \alpha} = \sum_{i=1}^{n} \frac{\gamma_i^t}{\alpha} - \frac{1-\gamma_i^t}{1-\alpha} = 0$$

$$\Rightarrow \frac{\sum_{i=1}^{n} \gamma_i^t}{\alpha} = \frac{n - \sum_{i=1}^{n} \gamma_i^t}{1-\alpha} \Rightarrow \sum_{i=1}^{n} \gamma_i^t \cancel{- \alpha \sum_{i=1}^{n} \gamma_i^t} = n\alpha \cancel{- \alpha \sum_{i=1}^{n} \gamma_i^t}$$

$$\Rightarrow \boxed{\hat{\alpha} = \frac{1}{n} \sum_{i=1}^{n} \gamma_i^t}$$

$$\frac{\partial Q}{\partial \lambda_1} = \sum_{i=1}^{n} \frac{\gamma_i^t}{\lambda_1} - x_i \gamma_i^t = 0 \Rightarrow \frac{\sum_{i=1}^{n} \gamma_i^t}{\lambda_1} = \sum_{i=1}^{n} \gamma_i^t x_i$$

$$\Rightarrow \boxed{\hat{\lambda}_1 = \frac{\sum_{i=1}^{n} \gamma_i^t}{\sum_{i=1}^{n} \gamma_i^t x_i}}$$

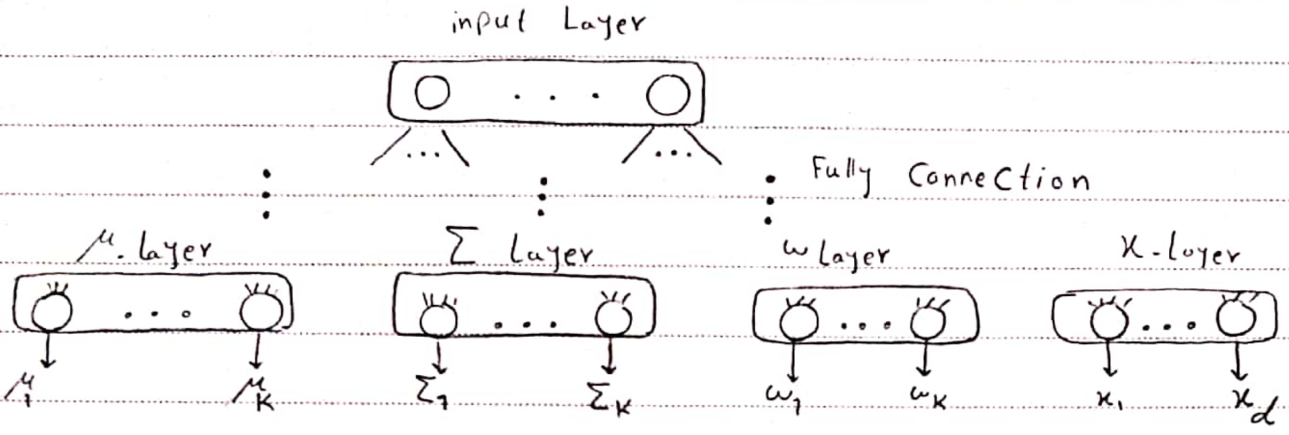
$$\frac{\partial Q}{\partial \lambda_2} = \sum_{i=1}^{n} (1-\gamma_i^t)\left(\frac{1}{\lambda_2} - x_i\right) = 0 \Rightarrow \frac{\sum_{i=1}^{n} (1-\gamma_i^t)}{\lambda_2} = \sum_{i=1}^{n} (1-\gamma_i^t) x_i$$

$$\Rightarrow \boxed{\hat{\lambda}_2 = \frac{\sum_{i=1}^{n} (1-\gamma_i^t)}{\sum_{i=1}^{n} (1-\gamma_i^t) x_i}} \quad \blacksquare$$

Subject:
Date

سوال ④ :

الف)



فرض می‌کنیم که توزیع از K تا توزیع گوسی تشکیل شده است

$$P(x) = \sum_{K=1}^{K} \omega_K \mathcal{N}(x, \mu_K, \Sigma_K)$$

که μ_K و Σ_K به ترتیب میانگین و قطر ماتریس Covariance می‌باشند ω_K میزان مشارکت توزیع K ام در توزیع نهایی را نشان می‌دهد، x-layer نیز همان ورودی است که به لایه‌ی آخر منتقل می‌شود البته d بعد آن به گونه‌ای train می‌شود که از هم مستقل باشند. به همین خاطر فقط قطر ماتریس Covariance را train کردیم و مابقی را صفر قرار می‌دهیم چون فرض می‌کنیم بعدهایی که بدست می‌آوریم مستقل خواهند بود.

PAPCO

Subject:
Date

ب) خروجی توابع فعال سازی را با علامت ' برایم نشان می دهیم:

* μ layer : هیچ محدودیتی برای وجود ندارد پس تابع فعال سازی قرار نمی دهیم

$\mu'_{ki} = \mu_{ki}$ (μ_K عنصر iام بردار)

* Σ-layer : همان طور که گفته شد پارامترهای ذخیره شده در این لایه معادل قطر ماتریس Covariance است، در نتیجه برای اینکه ویژگی ماتریس semi-positivity ماتریس Covariance حفظ شود از تابع فعال سازی نمایی استفاده می کنیم.

$\Sigma'_{ki} = \exp(\Sigma_{ki})$ (Σ_K عنصر iام بردار)

* w-layer : همه ی w_K ها باید مثبت باشند و مجموع آنها برابر یک شود.

در نتیجه می توانیم از Softmax استفاده کنیم که هر دو ویژگی گفته شده را داراست:

$$w'_K = \frac{\exp(w_K)}{\sum_{K=1}^{K} \exp(w_K)} \quad (\text{Softmax})$$

ج) می توانیم منفی log likelihood را به عنوان هزینه یا همان loss در نظر بگیریم (با پیروی از ایده Maximum likelihood):

$$Loss(x) = -\log \sum_{K=1}^{K} w_K N(x; \mu_K, \Sigma_K)$$

PAPCO

section

5 PCA

In this section, we will implement PCA step by step.

5.1 Read Dataset

First, load the fashion MNIST and plot an image of that:

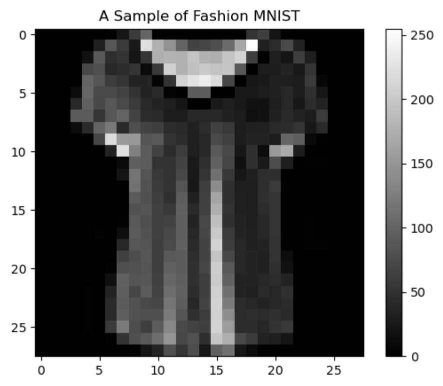


Figure 5.1: A sample image from fashion MNIST dataset

5.2 Standardization

We scale the data to have a mean of zero and a standard deviation of one. This is a necessary step of PCA because PCA assumes this scale during the mathematical calculations.

5.3 Covariance Matrix

Next, calculate the covariance matrix which has the shape of 784 by 784, where 784 is the number of features.

5.4 Eigenvalues & Eigenvectors

Now, Eigenvalues and eigenvectors of the covariance matrix should be extracted; The following figure shows eigenvalues in sorted representation.

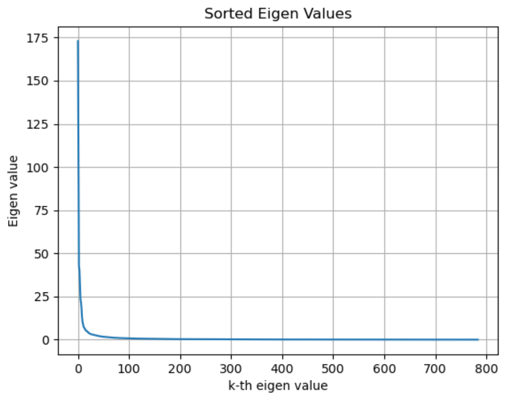


Figure 5.2: Sorted eigenvalues

A good way to find the optimal number of components in PCA is by looking at the eigenvalues. We plot them as above and apply the elbow method to do so. Let's zoom in on the figure for a better visualization and accurate estimation of the elbow method.

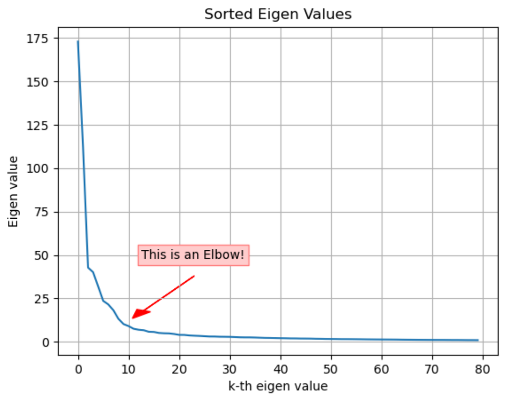


Figure 5.3: Find the optimal number of components with elbow method

As you can see in the figure above, using ten components for PCA can result in a good answer.

5.5 Compressing

So, we choose the 10 corresponding eigenvectors and create matrix W with a shape of 784 by 10. Now multiply the design matrix by W to compress data to 10 dimensions. The following shows a random image of the data set before and after compression.

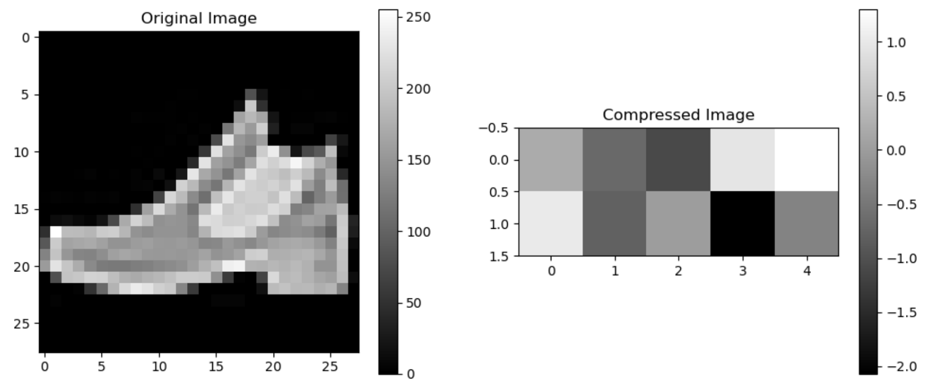


Figure 5.4: A sample of compressed images

6 GMM

In this section, we will discover the results of applying the Gaussian Mixture Model (GMM) to the MNIST dataset.

6.1 Fit GMM

First, we reduce the dimension of the data to 2 using PCA and then fit a GMM with two components.

6.2 Distance Between The Means

Euclidean distance between the means of each component is equal to 1909.64. We will explore the distance between the means more in the last subsection.

6.3 Visualizing Components Means

We bring the 2-dimensional means back to the original 784-dimensional space by applying the `inverse_transform` function. You can see these means in the following figures.

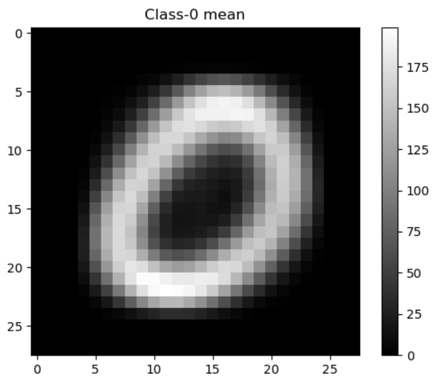


Figure 6.1: Class-0 mean in 784-dimensional space

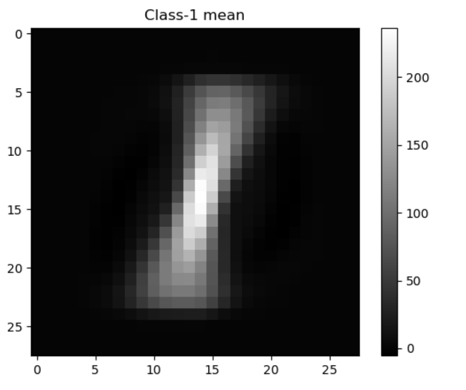


Figure 6.2: Class-1 mean in 784-dimensional space

As you can see in the figures, each one has a shaded number. This is because there were multiple handwritten examples of each number, and some of them were slightly rotated or had minor variations compared to others. Despite these small differences, the figures generally share common pixel patterns.

By taking the average of all the figures for a given number, the common pixels become more pronounced and appear whiter. This is because the pixels that were consistently close to the maximum value of 255 across the different examples now have intensity values nearer to 255 in the averaged figure.

On the other hand, the pixels that sometimes had values close to 255 and other times close to 0 (due to the variations in the handwritten examples) now have an average value of around 128, resulting in a gray appearance in the final figure. This averaging process helps to highlight the core pixel patterns that are shared among the multiple examples of each handwritten number.

6.4 Images Near the Decision Boundary

We calculate the difference in the probability of belonging to each class and find the corresponding images with the smallest values. The resulting images are as follows:

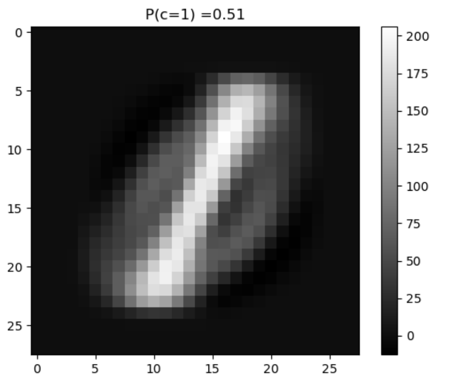


Figure 6.3: For this particular image, the differences in the probabilities across the possible classes are minimal.

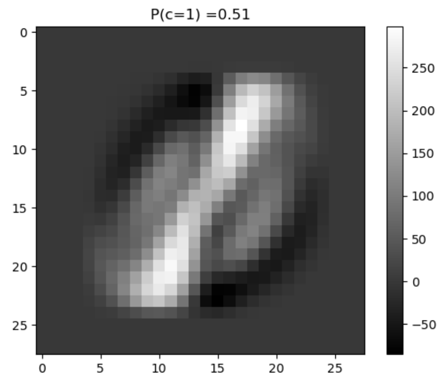


Figure 6.4: For this particular image, the differences in the probabilities across the possible classes are the second smallest.

As you can see, it is hard to determine which classes these samples belong to because they do not follow the dominant pattern in any of the classes. Furthermore, the probability of the samples belonging to each class is almost equal, indicating they lie on the decision boundary.

6.5 Similarity Between Classes

To analyze the similarities and differences between the handwritten digit classes, we performed the following steps:

1. For each pair of classes, We fit a Gaussian Mixture Model (GMM) with two components to the digit of these classes.
2. We calculated the distance between the means of the two GMM components.
3. Finally, we identified the pair of classes with the minimum distance between their GMM component means, as well as the pair with the maximum distance.

The results of this analysis showed that the class 0 and class 1 digits had the maximum distance between the means of their corresponding GMM components, as illustrated in Figures 6.1 and 6.2.

Conversely, the class 8 and class 9 digits exhibited the minimum distance between the means of their GMM components, as depicted in Figures 6.5 and 6.6.

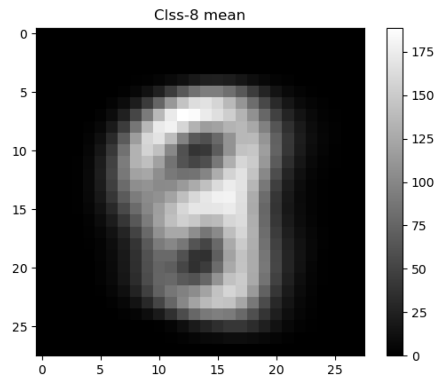


Figure 6.5: Class-8 mean in 784-dimensional space

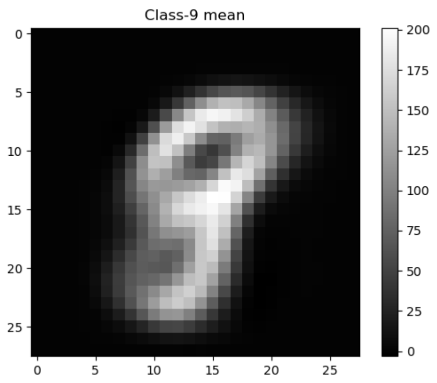


Figure 6.6: Class-9 mean in 784-dimensional space

The similarity between the means of class 8 and class 9 GMM components can be attributed to the fact that these two classes' digits generally share a significant number of common pixel patterns in their handwritten forms. This close resemblance is reflected in the proximity of their component means.

On the other hand, the class 0 and class 1 digits have fewer common pixel patterns, resulting in more distinct means for their GMM components. This dissimilarity suggests that these two classes' digits are more visually differentiated from one another.

7 Clustering

7.1 Definitions

1. **K-means Distortion and Elbow Method**: The K-means Distortion is a measure of the goodness of fit for a K-means clustering model. It is calculated as the sum of the squared distances between each observation vector and its assigned cluster centroid. The K-means Distortion represents the total within-cluster variance, and a lower distortion indicates a better clustering result.

   The Elbow Method is a graphical technique used to determine the optimal number of clusters (K) in a K-means clustering algorithm. The method involves plotting the K-means Distortion on the y-axis against the different values of K on the x-axis. The optimal K value is the point at which the graph forms an elbow.

   The logic behind the Elbow Method is that as the number of clusters increases, the K-means Distortion will decrease, but the rate of decrease will slow down after a certain point. The "elbow" on the plot corresponds to the point where adding more clusters does not significantly reduce the distortion, and this is usually considered the optimal number of clusters.

2. **Silhoutte Score**: It is a numerical value between -1 and 1 that indicates how well a data point belongs to its cluster and how separated it is from other clusters. A high silhouette score means that the data point is close to the average distance of its cluster and far from the nearest neighboring cluster, implying good clustering. Conversely, a low silhouette score means that the data points are far from the average distance of its cluster and close to another cluster, implying poor clustering. A silhouette score of 0 means that the data point is on the border of two clusters, indicating an ambiguous clustering.

   To calculate the silhouette score for a data point, we need to compute two values: a and b. The value a represents the average distance of the data point to all other data points in the same cluster. The value b represents the minimum average distance of the data point to all other data points in any other cluster. The silhouette score for the data point is then given by the formula:

$$S = \frac{b - a}{max(a, b)}$$

3. **Dacies-Bouldin Index**: The Davies-Bouldin Index is defined as the average similarity between each cluster and its most similar cluster. Mathematically, it can be expressed as:

$$DB = \frac{1}{k} \sum_{i} max_j(R_{i,j}))$$

where $R_{i,j}$ is the similarity measure between the i-th and j-th clusters, defined as:

$$R_{i,j} = \frac{s_i + s_j}{d(c_i, c_j)}$$

where s_i is the average distance of the points in the i-th cluster to the cluster centroid c_i. and $d(c_i, c_j)$ is the distance between the centroids of the i-th and j-th clusters.

4. **Calinski-Harabasz Index**: It is a metric for evaluating clustering algorithms, its formula is as follows:

$$CH = \frac{BCSS * (n - k)}{WCSS * (k - 1)}$$

where BCSS (Between-Cluster Sum of Squares) is the weighted sum of squared Euclidean distances between each cluster centroid (mean) and the overall data centroid (mean):

$$BCSS = \sum_{i=1}^{k} n_i ||c_i - c||^2$$

where n_i is the number of points in cluster C_i, c_i is the centroid of C_i, and c is the overall centroid of the data.

and WCSS (Within-Cluster Sum of Squares) is the sum of squared Euclidean distances between the data points and their respective cluster centroids:

$$WCSS = \sum_{i=1}^{k} \sum_{x \in C_i} ||x - c_i||^2$$

5. **Dunn Index**: It is defined as the ratio of the minimum inter-cluster distance to the maximum intra-cluster distance.

   The Dunn Index aims to identify compact and well-separated clusters. A higher Dunn Index value indicates better clustering.

7.2 Find Optimal Number of Clusters

The result of the elbow method is as follows:

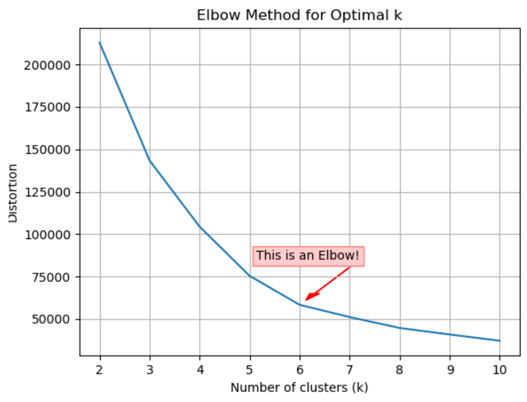


Figure 7.1: Elbow method for optimal k of k-means

As shown in the figure above, the optimal value of k is determined to be 6. Additionally, the four other methods discussed in the previous section also yielded the same optimal value of k, as demonstrated in the code accompanying this report.

7.3 Visualizing

We employ two techniques to visualize these clusters: Principal Component Analysis (PCA) and T-distributed Stochastic Neighbor Embedding (t-SNE).

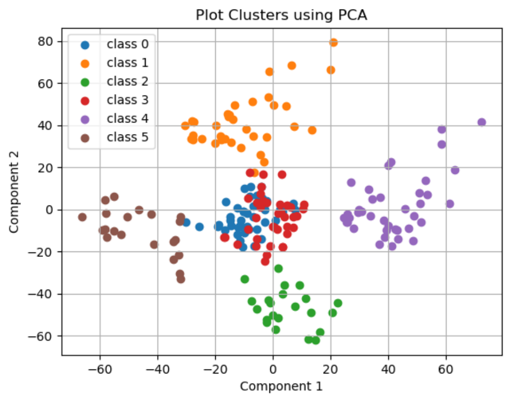


Figure 7.2: Visualizing clusters using PCA

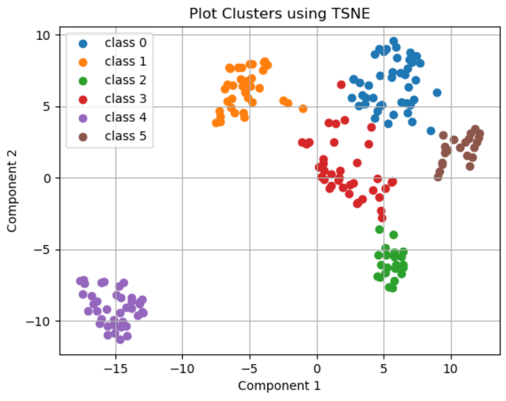


Figure 7.3: Visualizing clusters using t-SNE

t-SNE is a more effective technique for visualizing the clusters compared to PCA. This is because t-SNE aims to preserve the proximity of data points in the new, lower-dimensional space, such that samples that were close to each other in the original space remain close in the new representation, and vice versa. As evident in the figures, the clusters appear more compact in the t-SNE visualization, and the different clusters are also more distinctly separated from one another, compared to the PCA representation.