MATH38161 Multivariate Statistics and Machine Learning

Analysis of the FMNIST data using Principal Component Analysis and Gaussian Mixture Models

Brian Ezinwoke: 10827287

November 2024

Figure 1: First 15 images from our minimised FMNIST dataset

we

1 Dataset

The Fashion MNIST (FMNIST) dataset contains 70,000 grayscale images of fashion products categorised into 10 distinct classes. It is widely used as as benchmark in machine learning [1]. The dataset is strutured as follows:

- **Image Dimension**: Each image is 28x28 pixels, flattened into a vector of 784 pixel values.
- **Classes**: There are 10 distinct classes, each representing a type of clothing or footwear: T-shirt (0), Trouser (1), Pullover (2), Dress (3), Coat (4), Sandal (5), Shirt (6), Sneaker (7), Bag (8), Ankle Boot (9).
- **Dataset Size**:
  - Training set: 60,000 images
  - Test set: 10,000 images
- **Pixel Values**: Each pixel value ranges from 0 (black) to 255 (white).

In this project, a **subset containing 10,000 images of the full FMNIST dataset** has been analysed. A plot of the first 15 images can be seen in Figure 1.

2 Methods

2.1 Dimension reduction for FMNIST data using Principal Components Analysis (PCA)

Principal Component Analysis (PCA) is a statistical method used to reduce the dimensionality of data while retaining as much variability as possible. It achieves this by transforming the original variables into a new set of variables called principal components (PCs), which are uncorrelated and ordered by the amount of variance they capture from the data.

The FMNIST dataset consists of grayscale images of clothing items, with each image represented as a flattened vector of 784 pixel values. This high-dimensional data poses challenges for visualisation, interpretation, and computational efficiency, particularly for tasks like clustering. To address these challenges, we employed PCA for dimensionality reduction while retaining most of the dataset's variance.

2.1.1 Normalisation of Pixel Values

To ensure comparability among features, the pixel values were scaled from [0, 255] to a range of [0,1]. This step is critical for PCA, as unnormalised features can lead to dominance by variables with larger magnitudes.

```
# normalise pixel values
X <- (fmnist$x) / 255
```

Listing 1: Normalisation of Pixel Values

2.1.2 Computation of Principal Components (PCs)

PCA was applied to the normalised data to compute all 784 principal components. Each PC represents a linear combination of the original pixel values. The eigenvalues and eigenvectors of the covariance matrix of the dataset were calculated to determine the PCs and their corresponding variances.

```
# calculate principle components
fmnist_pca <- prcomp(X)
# calculate covariance of the principal components
fmnist_cov <- zapsmall(cov(fmnist_pca$x))
```

Listing 2: Computation of Principal Components

2.1.3 Proportion of Variance Explained (PVE)

The proportion of variance explained was calculated and the scree plot, Figure 2a, was generated to visualise the proportion of variance explained by each principal component. This is to help us select the most impactful components for analysis.

```
# calculate proportion of variance explained
PoVE <- (fmnist_pca$sdev^2) / sum(fmnist_pca$sdev^2)
plot(PoVE, type = "o", col = "blue",
 xlab = "Principal␣Component",
 ylab = "Proportion␣of␣Variance␣Explained",
 main = "Variance␣Explained␣by␣Each␣Principal␣Component")
```

Listing 3: Proportion of Variance Explained

2.1.4 Visualisation of Principal Components

A scatter plot of the first two principal components was created to visualise the structure of the data in the reduced space. The data points were coloured according to their true labels, illustrating how well the reduced dimensions captured separability among classes.

```
# visualise the first two principal components
plot(fmnist_pca$x[,1], fmnist_pca$x[,2],
    xlab = "PC1", ylab = "PC2",
    main = "Visualisation␣of␣Top␣2␣Principal␣Components",
    col = labels, pch =  19)
# add legend to bottom left
legend("bottomleft", legend = levels(labels),
    col = unique(labels), pch = 19,  title = "Labels")
```

Listing 4: Visualisation of Principal Components

2.1.5 Correlation Loadings Plot

A loadings plot was constructed to examine the contributions of the original variables (pixels) to the first two principal components. This plot provided insights into which regions of the image influenced the reduced components most significantly.

```r
# Correlation loadings
loadings <- cor(X, fmnist_pca$x) # calculate correlation loadings
# set up the plot
plot(1, xlim = c(-1, 1), ylim = c(-1, 1), type = "n,", asp = 1,
     xlab = "Correlation␣with␣PC1", ylab = "Correlation␣with␣PC2",
     main = "Correlation␣Loadings␣Plot␣for␣FMNIST␣Data")
# add circle
curve((sqrt(1 - x^2)), add = TRUE, from = -1, to = 1)
curve((-sqrt(1 - x^2)), add = TRUE, from = -1, to = 1)
# add axes
abline(h = 0)
abline(v = 0)
# add points
points(loadings[,  1], loadings[, 2], pch = 20, col = "red")
```

Listing 5: Correlation Loadings Analysis

2.2 Analysis of the FMNIST data set using Gaussian mixture models (GMMs)

Gaussian Mixture Models (GMMs) are a probabilistic clustering method used to model data as a mixture of multiple Gaussian distributions. Each distribution represents a cluster, and GMM assigns data points to clusters based on probabilities rather than hard assignments, as seen in methods like K-means.

Building upon the dimensionality reduction performed in Task 1, we analyzed the Fashion MNIST (FMNIST) dataset using GMMs to perform unsupervised clustering. GMMs are particularly useful in this analysis because they can handle overlapping clusters and adapt to varying cluster shapes, leveraging the lower-dimensional PCA representation of the data. Unlike hard clustering methods, GMMs allow for a probabilistic assignment of data points, providing a richer understanding of the class overlaps inherent in the FMNIST dataset. This is especially valuable for identifying substructures within classes, such as distinguishing between items with similar visual features, and for evaluating the intrinsic structure of the dataset without relying on predefined labels.

2.2.1 Data Preparation

Clustering directly on the original 784-dimensional data would be computationally infeasible and prone to noise. Therefore, we used the first 10 principal components derived in Task 1 as only they had a proportion of variance explained greater than 1%. This retained the majority of the dataset's variance while significantly reducing dimensionality.

```r
# select the first 10 principal components
pca_10 <- fmnist_pca$x[, 1:10]
```

Listing 6: Data Preparation

2.2.2 Model Fitting

GMMS with varying numbers of components were fitted to the data. We explored models with 2 to 15 components to determine the optimal number of cluster assigned by the model and to see how the model behaved with higher number of clusters [2].

```r
library(mclust)
# Compute the GMM with a range of 2:15 clusters
gmm_result <- Mclust(pca_10, G = 2:15)
```

```
# Output a summary of the results
summary(gmm_result)
```

Listing 7: Model Fitting

```
----------------------------------------------------------------------------
Gaussian finite mixture model fitted by EM algorithm
----------------------------------------------------------------------------

Mclust VVV (ellipsoidal, varying volume, shape, and orientation) model with 15
components:

 log-likelihood     n  df       BIC       ICL
      -109727.4 10000 989 -228563.8 -229915.8

Clustering table:
  1   2   3   4   5   6   7   8   9  10  11  12  13  14  15
733 881 438 689 582 760 643 988 678 656 660 750 384 488 670
```

2.2.3 Model Selection

The optimal number of clusters was determined using Bayesian Information Criterion (BIC). BIC balances model fit with model complexity typically favouring simple models that explain the data well. A plot of BIC values against the number of clusters was also generated to identify the elbow point of minimum, indicating the most appropriate number of classes.

```
plot(gmm_result, what = "BIC")
# Optimal number of clusters
optimal_clusters <- gmm_result$G
cat("Optimal␣number␣of␣clusters:", optimal_clusters, "\n")
```

Listing 8: Model Selection

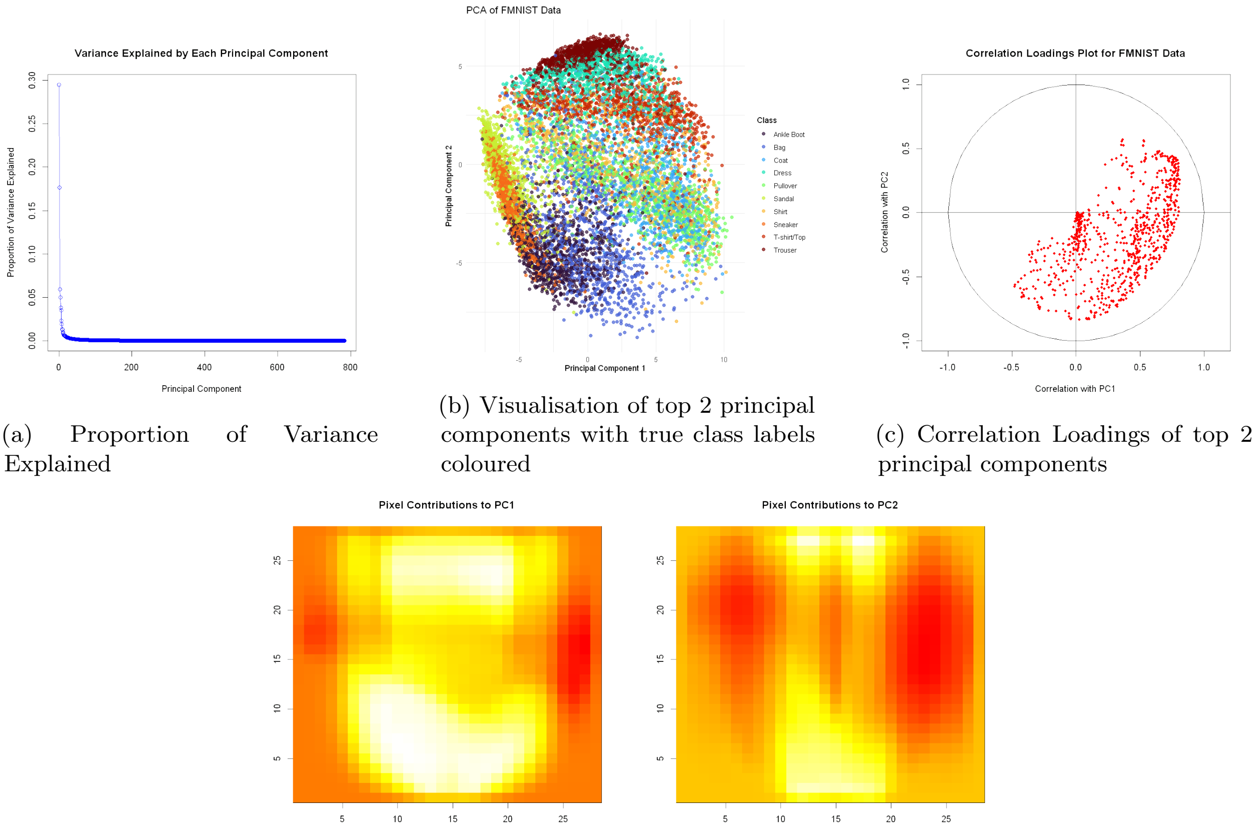
```
Optimal number of clusters: 15
```

2.2.4 Cluster Visualisation

The clustering results were visualised for each combination of reduced 2D PCA space. Data points were coloured based on their assigned cluster labels providing insights into the structure captured by the GMM. This structure was compared to the original labels to analyse the effectiveness of our GMM classifier [3].

```
library(ggplot2)
clusters <- gmm_result$classification

# Create a data frame for ggplot
plot_data <- data.frame(
  PC1 = pca_10[, 1],
  PC2 = pca_10[, 2],
  Cluster = factor(clusters),  # Clustering results
  Label = factor(labels)       # Optional: True labels
)

# Scatter plot with clusters as colours
ggplot(plot_data, aes(x = PC1, y = PC2, color = Cluster)) +
  geom_point(alpha = 0.7, size = 2) +  # Add points
  labs(title = "Clustering␣in␣PCA␣Space", x = "Principal␣Component␣1",
       y = "Principal␣Component␣2") +
  theme_classic() +
  scale_color_viridis_d(option = "H")  # Use viridis colours for distinct clusters

# Scatter plot with true labels as colours
ggplot(plot_data, aes(x = pca_10[, 1], y = pca_10[, 2], color = Label)) +
  geom_point(alpha = 1, size = 2) +  # Add points
  labs(title = "True␣Labels␣in␣PCA␣Space", x = "Principal␣Component␣1",
       y = "Principal␣Component␣2") +
  theme_minimal() +
  scale_color_viridis_d(option = "H")  # Use viridis colours for true labels

# Plots with all other combinations of components
plot(gmm_result, what = "classification") # clusters
plot(gmm_result, what = "uncertainty") # uncertainty
plot(gmm_result, what = "density") # density
```

Listing 9: Cluster Visualisation



(a) Proportion of Variance Explained

(b) Visualisation of top 2 principal components with true class labels coloured

(c) Correlation Loadings of top 2 principal components

(d) Heatmap of PC1

(e) Heatmap of PC2

Figure 2: Main Graphs for Dimension Reduction Using Principal Component Analysis: PoVE, Class Visualisation, Correlation Loadings, Heatmaps

3 Results and Discussion

3.1 Dimension Reduction using Principal Component Analysis

3.1.1 Effectiveness of the dimensionality reduction

PCA successfully reduced the dimensionality of the FMNIST dataset from 784 to 10 principal components while retaining most of the variability in the data. The scree plot, Figure 2a, showed that the first 10 components captured approximately 72% of the total variance, demonstrating that these components summarise the dataset well but still discard some information as noise.

3.1.2 Data Structure

The scatter plot of the first two principal components, Figure 2b, revealed some degree of clustering and separation among certain classes. Distinct groupings for items like Trousers and Sneakers were observed, whereas overlapping patterns were noted between visually similar items such as T-shirts and Shirts. This highlights both the effectiveness and limitations of PCA for capturing intrinsic structures in the data.

3.1.3 Loadings and Interpretability

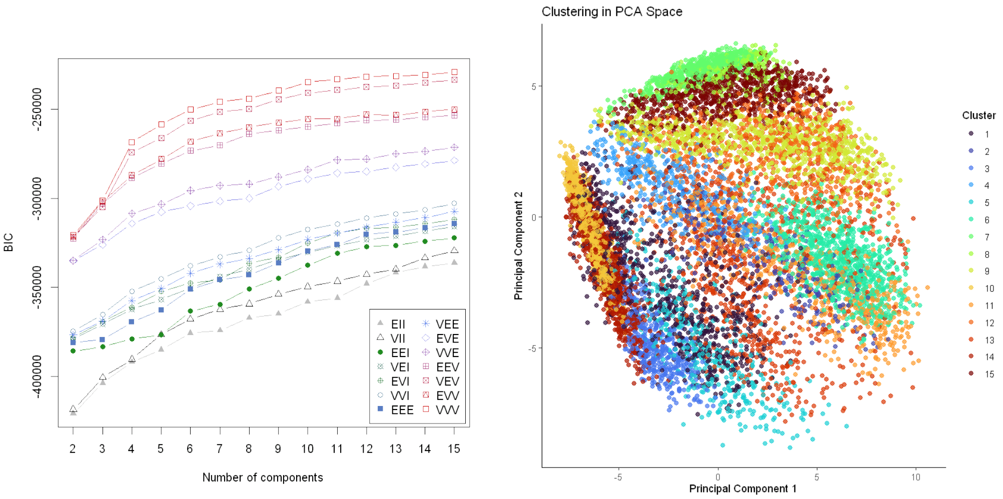
The correlation loadings plot, Figure 2c, indicated that the first principal component heavily weighs on pixel regions corresponding to key structural features (e.g., edges or contours of clothing). Similarly, the second component emphasises orthogonal variations such as patterns or secondary features. This is further supported by the heat maps of each principal component which can be seen in Figures 2d and 2e. These findings confirm that PCA captures meaningful patterns, but the principal components themselves lack semantic interpretability without further domain-specific knowledge.

```r
# Dimensions of the FMNIST images
img_width <- 28
img_height <- 28

# Get the loadings for the first principal component (PC1)
pc1_loadings <- fmnist_pca$rotation[, 1]  # Loadings for PC1

# Reshape into 28x28 matrix for visualisation
pc1_matrix <- matrix(pc1_loadings, nrow = img_height, ncol = img_width)

# Visualise as a heatmap
image(1:img_width, 1:img_height, pc1_matrix,
      col = heat.colors(256), xlab = "", ylab = "",
      main = "Pixel␣Contributions␣to␣PC1")

# Loadings for PC2
pc2_loadings <- fmnist_pca$rotation[, 2]
pc2_matrix <- matrix(pc2_loadings, nrow = img_height, ncol = img_width)

# Visualise as a heatmap
image(1:img_width, 1:img_height, pc2_matrix,
      col = heat.colors(256), xlab = "", ylab = "",
      main = "Pixel␣Contributions␣to␣PC2")
```

Listing 10: Heat map visualisations

3.1.4 Downstream Analysis

The reduction to 10 principal components drastically reduced computational complexity, making the data more manageable for clustering and classification tasks. This lower-dimensional representation provided a compact and robust foundation for subsequent analyses, such as Gaussian Mixture Models (GMMs).



(a) BIC by No of Components

(b) Clusters of Top 2 Components

Figure 3

3.1.5 Limitations

PCA is a linear method, so it may fail to capture nonlinear relationships in the data. Furthermore, the overlap between certain classes suggests that additional techniques (e.g., supervised learning or nonlinear dimensionality reduction) may be needed to improve separability.

3.2 GMM

3.2.1 Clustering Performance

The GMM fitted a VVV model with 15 components, which is out of alignment with the 10 categories in the FMNIST dataset. This is due to a high level of overlap between some clusters (e.g., T-shirts and pullovers), indicating that these items share similar visual features in the reduced PCA space. The GMM might have identified substructures within classes that are not labeled in the original dataset and hence some classes have been split into multiple cluster.

3.2.2 Effectiveness of BIC

Looking closer at the BIC plot, Figure 3a, we can see that we get better BIC scores as the number of components increases. Although this trend plateaus, this indicates that the BIC is not effective for this task as it favours more complex models and the correct cluster number, 10, is difficult to identify.

3.2.3 Impact of Dimensionality Reduction

Although the top 10 components retained 72% of the total variance, it is possible that critical information needed to distinguish certain classes led to them being split into multiple clusters. Selecting more components to explain more of the variance could fix this issue.

3.2.4 Visualisation and Interpretation

Clusters corresponding to distinctive classes (e.g., Trousers and Dresses) were well separated in the 2-D PCA scatter plot, Figure 3b, confirming that these items have distinct visual features. In contrast, classes that are harder to distinguish due to shared structural characteristics have been separated into smaller components. Further inspection of other components can be seen in Figure 4 which supports the notion that there is a high level of overlap within clusters.

3.2.5 Misclassification Rate

The misclassification rate of 0.5226 is very high showing that our 15 cluster model does not align well with the true labels in the FMNIST dataset. This is due to reasons already mentioned: overlapping classes, suboptimal number of clusters selected using the BIC and the limitations of PCA which meant that critical information may have been lost.

However, when we set our cluster number to be 10, we see that our misclassification rate reduces to 0.3884. This shows that with the true number of classes, our model is better aligned as it reduces the splitting of clusters into multiple clusters. Additionally, the 15 cluster model likely overfit to the data which was causing the increased error. Regardless, the misclassification rate is still high, demonstrating the challenge in separating the FMNIST classes with a GMM.

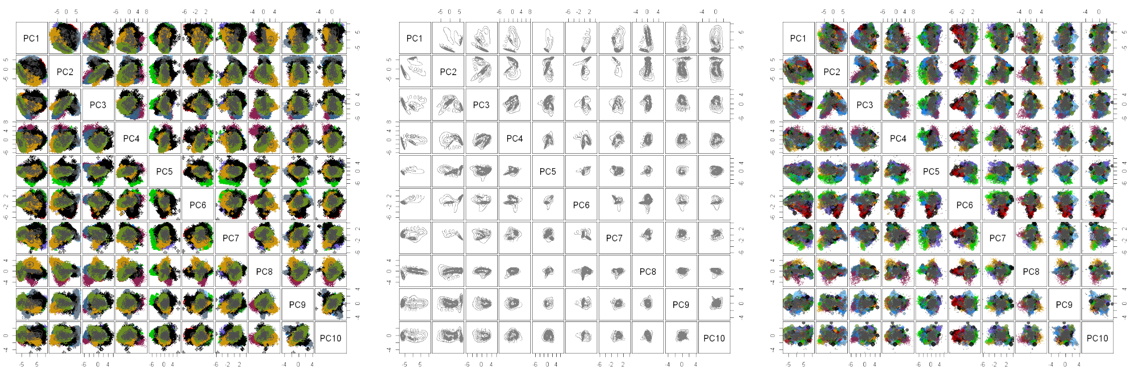
```r
# Create a confusion matrix comparing true vs predicted labels
confusion_matrix <- table(labels, gmm_result$classification)

# Calculate the number of correctly classified points
correct_classifications <- sum(apply(confusion_matrix, 1, max))

# Total number of points
total_points <- sum(confusion_matrix)

# Calculate the misclassification rate
misclassification_rate <- 1 - (correct_classifications / total_points)
print(paste("15␣Clusters␣Misclassification␣Rate:",
            round(misclassification_rate, 4)))

# Compute the GMM with 10 clusters
gmm_10 <- Mclust(pca_10, G = 10)

# Create a confusion matrix
conf_mat10 <- table(labels, gmm_10$classification)

# Calculate the number of correctly classified points
correct_classifications <- sum(apply(conf_mat10, 1, max))

# Total number of points
total_points <- sum(conf_mat10)

# Calculate the misclassification rate
misclassification_rate <- 1 - (correct_classifications / total_points)
print(paste("10␣Clusters␣Misclassification␣Rate:",
            round(misclassification_rate, 4)))
```

Listing 11: Confusion matrix and misclassification rate of GMM

```
[1] "15 Clusters Misclassification Rate: 0.5226"

[2] "10 Clusters Misclassification Rate: 0.3884"
```

3.2.6 Limitations

GMM assumes Gaussian clusters, which may not accurately represent the complex structures of FMNIST categories. Furthermore, dimensionality reduction via PCA, while effective, may discard class-specific nuances. Moreover, cluster number selection using BIC can be misleading so it is best to test and compare error rates to find the true cluster number.



(a) Cluster scatter plot

(b) Density Plot

(c) Uncertainty Plot

Figure 4: Classification, Density and Uncertainty Plots for all Principal Components

4 Conclusion

4.1 Summary

In this analysis, we utilised Principal Component Analysis (PCA) to reduce the dimensionality of the FMNIST dataset from 784 to 10 components, retaining 72% of the variance while highlighting meaningful patterns in the data, such as edges and contours. However, overlapping class features, particularly among similar items like T-shirts and Shirts, indicated the limitations of PCA as a linear method. Gaussian Mixture Models (GMMs) were then employed for clustering in the reduced PCA space. While the model initially identified 15 clusters, leading to a high misclassification rate of 52.26%, adjusting the cluster count to match the true number of classes (10) reduced the error to 38.84%. This improvement highlights the importance of aligning cluster counts with dataset structure but also underscores persistent challenges due to overlapping classes, loss of critical information in dimensionality reduction, and GMM's Gaussian assumptions. Overall, while PCA and GMM provided valuable insights into the FMNIST data structure, the results reveal the need for more advanced non-linear dimensionality reduction and clustering techniques to achieve greater separability and interpretability for such complex datasets.

4.2 Future Work

- Explore non-linear methods like t-SNE or UMAP to better preserve class-specific structures during dimensionality reduction.
- Experiment with alternative clustering techniques, such as hierarchical clustering or density-based methods like DBSCAN, which may better handle overlapping classes.
- Implement deep learning techniques that combine feature extraction and clustering in a unified framework to capture complex patterns.
- Instead of solely relying on BIC, consider external metrics (e.g., Adjusted Rand Index) to evaluate cluster quality relative to true labels.
- Conduct further analysis of the correlation loadings to identify specific pixel regions associated with key visual features and link the results back to domain knowledge. [4]

References

[1] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017.

[2] Luca Scrucca, Chris Fraley, T. Brendan Murphy, and Adrian E. Raftery. *Model-Based Clustering, Classification, and Density Estimation Using mclust in R.* Chapman and Hall/CRC, 2023.

[3] Hadley Wickham. *ggplot2: Elegant Graphics for Data Analysis.* Springer-Verlag New York, 2016.

[4] OpenAI. Chat-gpt4, 2024.