Fashion MNIST Dataset Classification K-Means Only, K-Means and GMM Using Auto Encoder.

Sunnam Ravikiran
Department of Computer Science
University at buffalo
buffalo, NY 14221
rsunnam@buffalo.edu

Abstract

Unsupervised algorithms make inferences from datasets using only input vectors without referring to known, or labelled, outcomes. K-Means, GMM techniques have been applied on the MINST dataset to predicted clusters. The same has a been applied on the compressed images using auto encoder. The data set is split into two portions on which model is trained, validated and tested. It was observed that the model predicted the classes with good accuracy.

1 Introduction

Unsupervised learning is a type of machine learning algorithm used to draw inferences from datasets consisting of input data without labeled responses.

The most common unsupervised learning method is cluster analysis, which is used for exploratory data analysis to find hidden patterns or grouping in data. The clusters are modeled using a measure of similarity which is defined upon metrics such as Euclidean or probabilistic distance.

Common clustering algorithms include:

- **Hierarchical clustering:** builds a multilevel hierarchy of clusters by creating a cluster tree
- **k-Means clustering:** partitions data into k distinct clusters based on distance to the centroid of a cluster
- **Gaussian mixture models:** models clusters as a mixture of multivariate normal density components
- **Self-organizing maps:** uses neural networks that learn the topology and distribution of the data
- **Hidden Markov models:** uses observed data to recover the sequence of states

An **autoencoder** is a type of artificial neural network used to learn efficient data codings in an unsupervised manner. The aim of an autoencoder is to learn a representation (encoding) for a set of data, typically for dimensionality reduction, by training the network to ignore signal "noise". Along with the reduction side, a reconstructing side is learnt, where the autoencoder tries to generate from the reduced encoding a representation as close as possible to its original input.

2 Dataset

For training and testing of our clustering techniques, we will use the Fashion-MNIST dataset. The Fashion-MNIST is a dataset of Zalando's article images, consisting of a training set of 60,000 examples and a test set of 10,000 examples. Each example is a 28x28 grayscale image, associated with a label from 10 classes. Each image is 28 pixels in height and 28 pixels in width, for a total of 784 pixels in total. Each pixel has a single pixel-value associated with it, indicating the lightness or darkness of that pixel, with higher numbers meaning darker. This pixel-value is an integer between 0 and 255. The training and test data sets have 785 columns. The first column consists of the class labels (see above), and represents the article of clothing. The rest of the columns contain the pixel-values of the associated image.

1	T-shirt/top
2	Trouser
3	Pullover
4	Dress
5	Coat
6	Sandal
7	Shirt
8	Sneaker
9	Bag
10	Ankle Boot

Table 1: Labels for Fashion-MNIST dataset

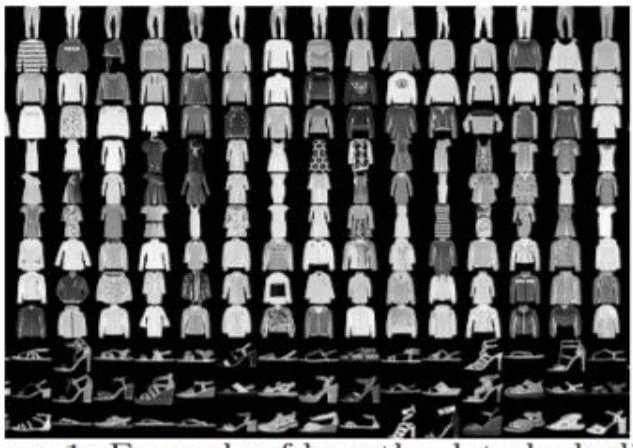


Figure 1: Example of how the data looks like.

3 Preprocessing the dataset

The given Fashion MNIST dataset has been preprocessed before using logistic regression model as follows.

- *Processing the data*
    1. The dependent valuable is identified by reshaped accordingly, matching the size of the out network.
    2. The independent variables which don't contribute the prediction of the dependent variable are dropped from the dataset.

- *Data Partitioning*
  In machine learning we usually split our data into three subsets: train, validate and test dataset, and fit our model on the train data, in order to make predictions on the test data. This is done to avoid the overfitting of the model.

  The Given dataset is partitioned into training, validation and testing data. Randomly data is split where training dataset has 60000 instances and test dataset has 10000 instances.

  ```
  1. The training dataset contains [60000] instances.
  2. The validation dataset contains [10000] instances.
  3. The testing dataset contains [10000] instances.
  4.
  ```

- *Normalization*
  Feature scaling is a method used to normalize the range of independent variables or features of data. In data processing, it is also known as data normalization and is generally performed during the data preprocessing step. The goal of normalization is to change the values of numeric columns in the dataset to a common scale, without distorting differences in the ranges of values. All the datasets (train, validation and test) have been mean normalized.

4 Architecture

K- Means

K-means clustering is one of the simplest and popular unsupervised machine learning algorithms. The objective of K-means is simple: group similar data points together and discover underlying patterns. To achieve this objective, K-means looks for a fixed number (k) of clusters in a dataset.

A cluster refers to a collection of data points aggregated together because of certain similarities.

the K-means algorithm identifies k number of centroids, and then allocates every data point to the nearest cluster, while keeping the centroids as small as possible.

- **Initialization:**
  Firstly, you need to randomly initialize two points called the cluster centroids. Here, you need to make sure that your cluster centroids depicted by an orange and blue cross as shown in the image are less than the training data points depicted by navy blue dots. k-means clustering algorithm is an iterative algorithm and it follows next two
- **Cluster Assignment:**
  In this step, it will go through all the data points to compute the distance between the data points and the cluster centroid initialized in the previous step. Now, depending upon the minimum distance from the orange cluster centroid or blue cluster centroid, it will group itself into that particular group. So, data points are divided into two groups, one represented by orange color and the other one in blue color as shown in the graph. Since these cluster formations are not the optimized clusters, so let's move ahead and see how to get final clusters.

- **Move Centroid:**
  Now, you will take the above two cluster centroids and iteratively reposition them for optimization. You will take all blue dots, compute their average and move current cluster centroid to this new location. Similarly, you will move orange cluster centroid to the average of orange data points. Therefore, the new cluster centroids will look as shown in the graph. Moving forward, let's see how can we optimize clusters which will give us better insight.
- **Optimization:**
  You need to repeat above two steps iteratively till the cluster centroids stop changing their positions and become static. Once the clusters become static, then k-means clustering algorithm is said to be converged.

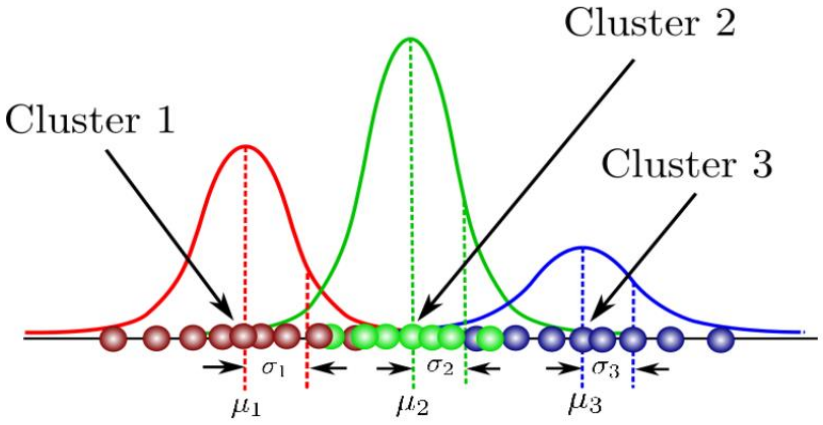
Given a set of observations (x1, x2, ..., xn), where each observation is a d-dimensional real vector, k-means clustering aims to partition the n observations into k (≤ n)
sets S = {S1, S2, ..., Sk} so as to minimize the within-cluster sum of squares (WCSS)
(i.e. variance). Formally, the objective is to find:

$$\arg\min_{\mathbf{S}} \sum_{i=1}^{k} \sum_{\mathbf{x} \in S_i} \|\mathbf{x} - \boldsymbol{\mu}_i\|^2 = \arg\min_{\mathbf{S}} \sum_{i=1}^{k} |S_i| \operatorname{Var} S_i$$

Gaussian mixture models

A Gaussian Mixture is a function that is comprised of several Gaussians, each identified by k ∈ {1…, K}, where K is the number of clusters of our dataset. Each Gaussian k in the mixture is comprised of the following parameters:

- A mean μ that defines its center.
- A covariance Σ that defines its width. This would be equivalent to the dimensions of an ellipsoid in a multivariate scenario.
- A mixing probability π that defines how big or small the Gaussian function will be.



we can see that there are three Gaussian functions, hence $K = 3$. Each Gaussian explains the data contained in each of the three clusters available. The mixing coefficients are themselves probabilities and must meet this condition:

$$\sum_{k=1}^{K} \pi_k = 1 \qquad (1)$$

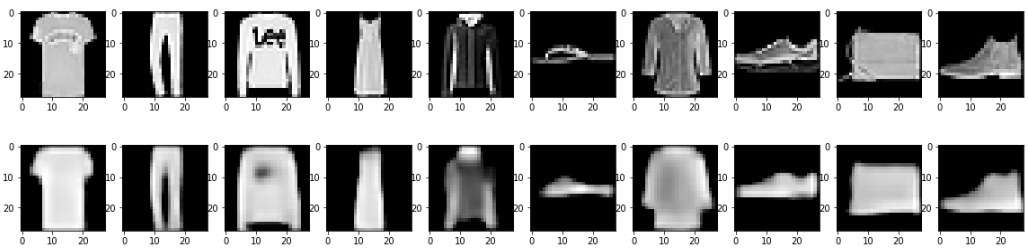
In general, the Gaussian density function is given by:

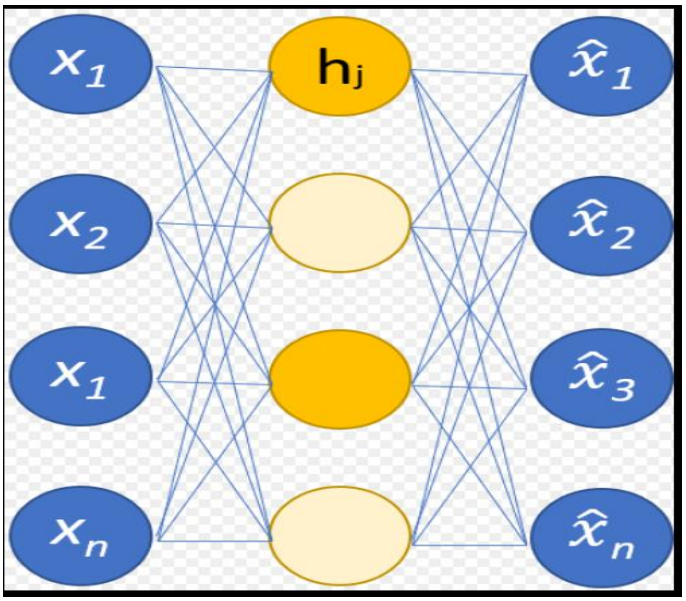
$$\mathcal{N}(\mathbf{x}|\mu, \Sigma) = \frac{1}{(2\pi)^{D/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\mu)^T\Sigma^{-1}(\mathbf{x}-\mu)\right)$$

Autoencoder

The idea of autoencoders has been popular in the field of neural networks for decades. Their most traditional application was dimensionality reduction or feature learning, but more recently the autoencoder concept has become more widely used for learning generative models of data. Some of the most powerful AIs in the 2010s involved sparse autoencoders stacked inside of deep neural networks.

The simplest form of an autoencoder is a feedforward, non-recurrent neural network similar to single layer perceptrons that participate in multilayer perceptrons (MLP) – having an input layer, an output layer and one or more hidden layers connecting them – where the output layer has the same number of nodes (neurons) as the input layer, and with the purpose of reconstructing its inputs (minimizing the difference between the input and the output) instead of predicting the target value Y and given input X.



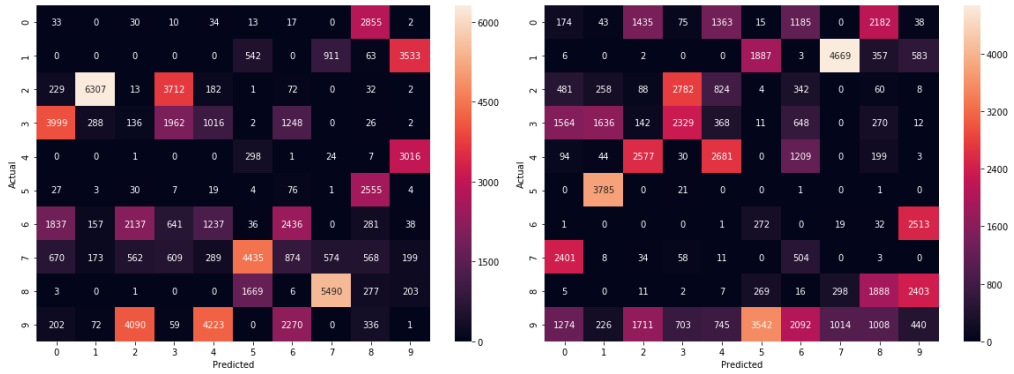


4.4 Results

Accuracy:

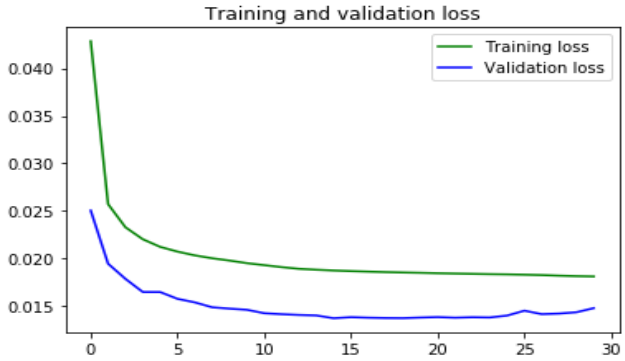
K-Means Only:	56.4%
Auto-Encoder based K-Means:	49.1%
Auto-Encoder based GMM:	54%

Confusion Matrix:
A confusion matrix is a table that is often used to describe the performance of a classification model (or "classifier") on a set of test data for which the true values are known. The confusion matrix itself is relatively simple to understand, but the related terminology can be confusing.

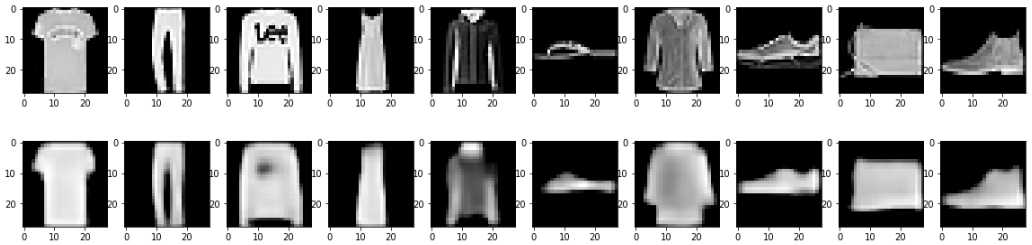


K-Means Only

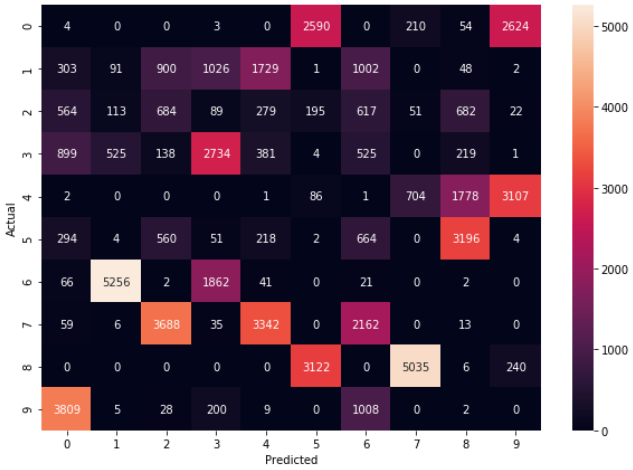
Auto-Encoder based K-Means



Training and validation loss for auto-encoder



Compressed Images using Autoencoder



Auto-Encoder based GMM

5 Conclusion

K-means clustering and Gaussian Mixture are one of the simplest and popular unsupervised machine learning algorithms. They prove to be very effective in clustering the data. In the above example K-Means and GMM were used to cluster data into 10 clusters successfully. An autoencoder is a type of artificial neural network used to learn efficient data codings in an unsupervised manner. Using autoencoder we have successfully compressed the image and applied K-means and GMM on the compressed image.