Unsupervised learning - Clustering

Christos Papadopoulos

December 2022

Contents

Chapter 1

Introduction

Clustering methods are very important techniques for exploratory data analysis with wide applications ranging from data mining [1, 2], dimension reduction, segmentation and so on. Their aim is to partition data points into clusters so that data in the same cluster are similar to each other while data in different clusters are dissimilar. Approaches to achieve this aim include partitional methods such as k-means and k-medoids, hierarchical methods like agglomerative clustering and divisive clustering, methods based on density estimation such as DBSCAN , and recent methods based on finding density peaks such as CFSFDP.

Image clustering is a special case of clustering analysis that seeks to find compact, object-level models from many unlabeled images. Its applications include automatic visual concept discovery [9], content-based image retrieval and image annotation. However, image clustering is a hard task mainly owning to the following two reasons: 1) images often are of high dimensionality, which will significantly affect the performance of clustering methods.To adress these issues ,recently, many approaches have been proposed to combine clustering methods with deep neural networks (DNN), which have shown a remarkable performance improvement over hand-crafted features., a kind of deep (convolutional) neural networks, such as deep belief network (DBN) and stacked auto-encoders , is first trained in an unsupervised manner to approximate the non-linear feature embedding from the raw image space to the embedded feature space (usually being low-dimensional). And then, either k-means or spectral clustering or agglomerative clustering can be applied to partition the feature space.However, since the feature learning and clustering are separated from each other, the learned DNN features may not be reliable for clustering.

In this paper we will develop combined deep learning models and clustering techniques with and without autoencoders, which will be used on the data of the fashion-mnist data set.The remaining of this paper includes the following: section 2 and 3 presents a brief description on autoencoders and the perforamnance metrics used, section 4, describes the experimental set up, section 5 presents the dataset used, section 6 presents the results and section 7 concludes the paper.

Chapter 2

Autoencoders

An autoencoder is actually an Artificial Neural Network that is used to decompress and compress the input data provided in an unsupervised manner. Decompression and compression operations are lossy and data-specific.
Data specific means that the autoencoder will only be able to actually compress the data on which it has been trained. For example, if you train an autoencoder with images of dogs, then it will give a bad performance for cats. The autoencoder plans to learn the representation which is known as the encoding for a whole set of data. This can result in the reduction of the dimensionality by the training network. The reconstruction part is also learned with this.
Lossy operations mean that the reconstructed image is often not an as sharp or high resolution in quality as the original one and the difference is greater for reconstructions with a greater loss and this is known as a lossy operation. The following image shows how the image is encoded and decoded with a certain loss factor.
The Autoencoder is a particular type of feed-forward neural network and the input should be similar to the output. Hence we would need an encoding method, loss function, and a decoding method. The end goal is to perfectly replicate the input with minimum loss.

2.1 Stacked Autoencoders

Some datasets have a complex relationship within the features. Thus, using only one Autoencoder is not sufficient. A single Autoencoder might be unable to reduce the dimensionality of the input features. Therefore for such use cases, we use stacked autoencoders. The stacked autoencoders are, as the name suggests, multiple encoders stacked on top of one another.

2.2 Denoising Autoencoders

Denoising Autoencodersy corrupt the data on purpose by randomly turning some of the input values to zero. In general, the percentage of input nodes which are being set to zero is about 50%. Other sources suggest a lower count, such as 30%. It depends on the amount of data and input nodes you have.
When calculating the Loss function, it is important to compare the output values with the original input, not with the corrupted input. That way, the risk of learning the identity function instead of extracting features is eliminated.

Chapter 3

Performance Metrics-Clustering

While Classification and Regression tasks form what's called Supervised Learning, Clustering forms the majority of Unsupervised Learning tasks. The difference between these two macro-areas lies in the type of data used. While in Supervised Learning samples are labelled with either a categorical label (Classification) or a numerical value (Regression), in Unsupervised Learning samples are not labelled, making it a relatively complex task to perform and evaluate.Correctly measuring the performance of Clustering algorithms is key. This is especially true as it often happens that clusters are manually and qualitatively inspected to determine whether the results are meaningful.In this chapter , we will go through the main metrics used to evaluate the performance of Clustering algorithms, to rigorously have a set of measures.

3.1 Silhouette Score

The Silhouette Score is used to measure the separation distance between clusters. It displays a measure of how close each point in a cluster is to points in the neighbouring clusters. This measure has a range of [-1, 1] and is a great tool to visually inspect the similarities within clusters and differences across clusters.
The Silhouette Score is calculated using the mean intra-cluster distance (i) and the mean nearest-cluster distance (n) for each sample. The Silhouette Coefficient for a sample is (n - i) / max(i, n). n is the distance between each sample and the nearest cluster that the sample is not a part of while i is the mean distance within each cluster.
The higher the Silhouette Coefficients (the closer to +1), the further away the cluster's samples are from the neighbouring clusters samples. A value of 0 indicates that the sample is on or very close to the decision boundary between two neighbouring clusters. Negative values, instead, indicate that those samples might have been assigned to the wrong cluster. Averaging the Silhouette Coef-

ficients, we can get to a global Silhouette Score which can be used to describe the entire population's performance with a single value.

3.2 Calinski-Harabasz Index

Calinski-Harabasz Index is also known as the Variance Ratio Criterion. The score is defined as the ratio between the within-cluster dispersion and the between-cluster dispersion. The C-H Index is a great way to evaluate the performance of a Clustering algorithm as it does not require information on the ground truth labels. The higher the Index, the better the performance.

3.3 Davies-Bouldin Index

The Davies-Bouldin Index is defined as the average similarity measure of each cluster with its most similar cluster. Similarity is the ratio of within-cluster distances to between-cluster distances. In this way, clusters which are farther apart and less dispersed will lead to a better score.The minimum score is zero, and differently from most performance metrics, the lower values the better clustering performance.Similarly to the Silhouette Score, the D-B Index does not require the a-priori knowledge of the ground-truth labels, but has a simpler implementation in terms of fomulation than Silhouette Score.

3.4 V measure

The V-measure is the harmonic mean between homogeneity and completeness.This metric is independent of the absolute values of the labels: a permutation of the class or cluster label values won't change the score value in any way. This metric is furthermore symmetric: switching true labels with labels predicted will return the same score value. This can be useful to measure the agreement of two independent label assignments strategies on the same dataset when the real ground truth is not known.The v measure score lies between 0.0 and 1.0. while 1.0 stands for perfectly complete labeling.

Chapter 4

Experimental SetUp

For the experiments we used fashion-mnist dataset which was splitted to train and test and then we split the train set to train and validate set with 90-10%ratio.Then we normalized pixels values between 0 and 1.After that , we performed three experiments.The first experiment was performed without autoencoder , the clustering technique directly uses the values of pixels scaled , the second with a stacked autoencoder and finally the third with a denoising autoencoder.Then we performed 3 clustering tecniques (kmeans-MiniBatch , Spectral Clustering, Agglomerative) and we compared the results of the performance metrics of each one.

Chapter 5

Dataset

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. We intend Fashion-MNIST to serve as a direct drop-in replacement for the original MNIST dataset for benchmarking machine learning algorithms. It shares the same image size and structure of training and testing splits. Each training and test example is assigned to one of the following labels of fig 1.1 ,while in fig 1.2 an example of the dataset for each class is given.

Description	Label
T-shirt/top	0
Trouser	1
Pullover	2
Dress	3
Coat	4
Sandal	5
Shirt	6
Sneaker	7
Bag	8
Ankle boot	9

Figure 5.1: Description per label

Figure 5.2: Example of the dataset for each class

Chapter 6

Results

Given we know that our dataset has 10 classes ,the metrics scores were all performed and compared for clustering into 10 clusters.

	Silhouette Score	V measure	Calinski-Harabasz Index	Davies-Bouldin Index
K-means MiniBatch	-0,13	0,49	1630,34	6,28
Spectral clustering				
Agglomerative clustering	-0,11	0,53	2060,93	8,38

Figure 6.1: Clustering scores for 10 clusters using the pixel values of the images (normalized to [0,1])

As Fig. 6.1 the spectral clustering did not work with the cause of memory in the case of using the pixel values of the images.Furthermore the Agglomerative clustering seem to perform better that the MiniBtach.

	Silhouette Score	V measure	Calinski-Harabasz Index ▾	Davies-Bouldin Index
Agglomerative clustering	-0,05	0,56	2142,74	17,87
K-means MiniBatch	-0,06	0,49	1700,57	8,08
Spectral clustering	-0,21	0,43	1141,22	8,9

Figure 6.2: Clustering scores for 10 clusters using the values of the images produced by the encoder

In fig. 6.2 most of the metrics used suggest that k-means MiniBatch and agglomerative clustering perfomed better than spectral clustering.

As fig. 6.3 suggests in our dataset the use of denoising encoder did not improve

	Silhouette Score	V measure	Calinski-Harabasz Index	Davies-Bouldin Index
K-means MiniBatch	-0,18	0,44	1525	9,59
Spectral clustering	-0,15	0,48	1546	5,71
Agglomerative clustering	-0,11	0,5	1732	15,13

Figure 6.3: Clustering scores for 10 clusters using the values of the images produced by the denoising encoder

the scores except of the Spectral clustering.The scores of the three clustering techniques are very close to draw safe conclusions for that experiment.

Chapter 7

Conclusion

In this paper we have implemented three different clustering techniques using different values of the images.Based on the results , we observed that, for our dataset using the values of the images produced by the encoder performed better and quicker than using the pixel values of the images, with agglomerative clustering performing sligtly better than MiniBatch.The use of the denoising encoder didnt make a significant imporovement in the scores.