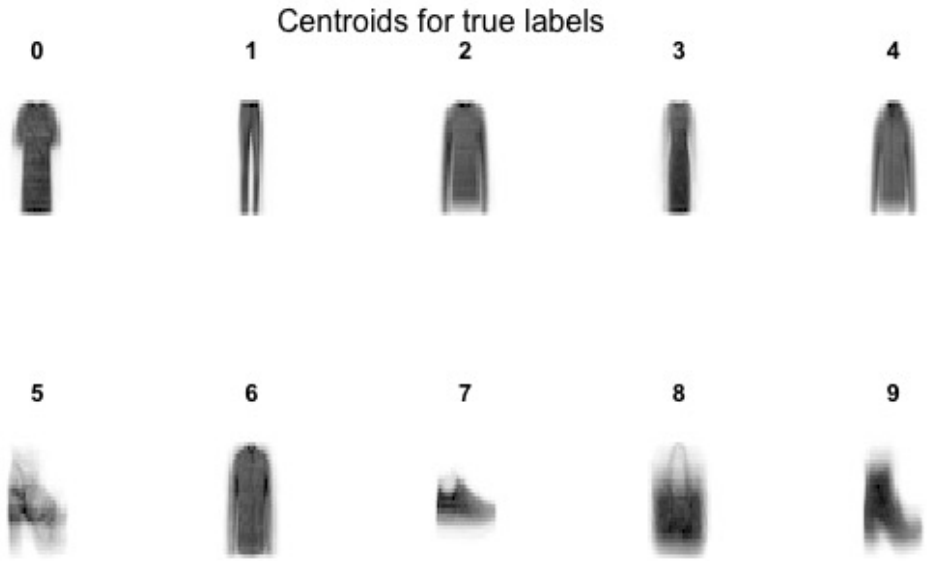
MSA220:NMF Clustering

May 2019

Introduction

- Using a subset of the high-dimensional fashion-MNIST dataset.
- The dataset consists of 10 different labels where each label represents a category of clothing.
- Each observation in the set is a 28x28 grayscale image associated with a label.
- Randomly selected 800 observations.

Introduction



- The figure shows the mean (centroid) of each label plotted as images.
- The higher the "blurriness" the higher the variance of the observations in a label.

NMF Clustering vs k-Means Clustering

- Compairing NMF clustering and k-means clustering to find the same number of clusters.
- Running both methods 10 times, with rank/clusters $= 10$.
- For each run, calculating Purity and Entropy as quality measures.
- Purity is a measure of cluster "quality" and to which extent a cluster contains a single lable (0 bad, 1 is perfect).
- Entropy is also a measure of cluster "quality" but measures the amount of disorder in a cluster. Small values indicates less disorder, which means better clustering.

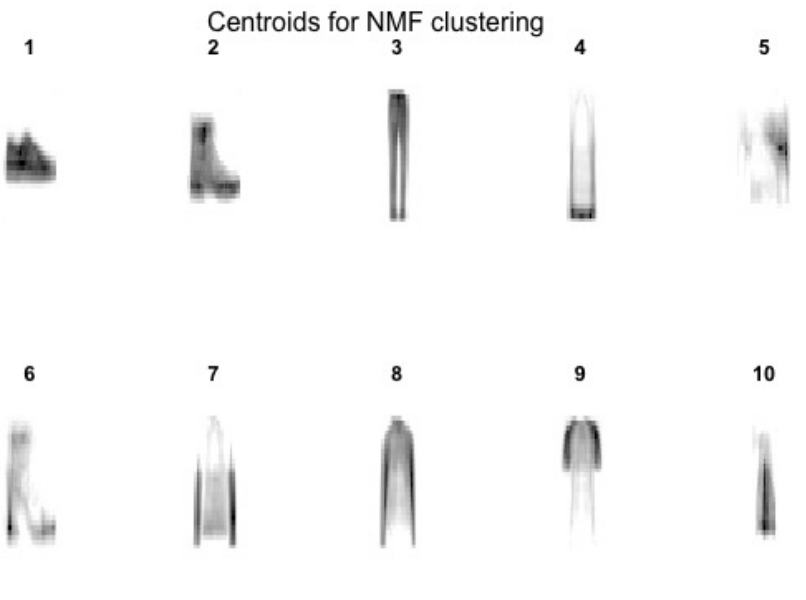
NMF Clustering vs k-means Clustering

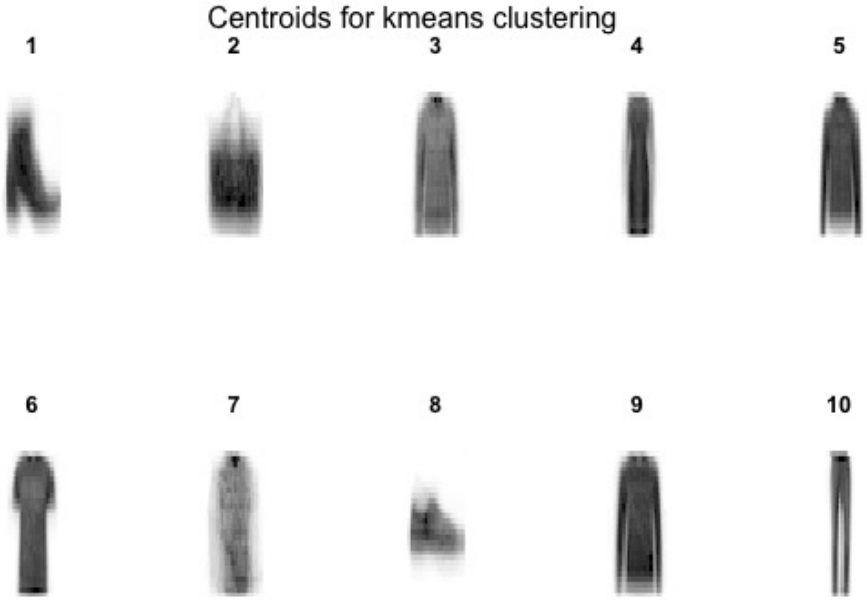
	NMF	k-means
Purity	0.51	0.52
Entropy	0.50	0.50

Table: Mean Purity and Entropy

- The table shows mean purity and entropy for NMF and k-means over the 10 runs.
- Both methods show similar results.

Comparing Cluster Centroids





- K-means seems to produce more accurate representations of the original images.
- Labels 4,5 and 10 (maybe more) are pretty much impossible to interpret for NMF.
- However, kmean seems to entirely ignore one of the shoes, as there were three types of shoes originally. It seems like it added a type of shirt instead. Also label 7 is hard to interpret.

Which labels ended up in which clusters?

Pred/Ref	0	1	2	3	4	5	6	7	8	9
0	0	0	0	0	0	37	0	60	2	5
1	0	0	0	0	0	21	0	9	6	48
2	7	79	0	64	9	0	2	0	0	0
3	7	0	1	8	1	0	6	0	8	0
4	1	0	6	1	0	2	8	0	25	0
5	0	0	4	0	0	15	3	0	4	19
6	0	0	31	1	12	0	19	0	24	0
7	0	1	35	6	64	0	31	0	0	0
8	60	0	15	2	0	0	13	0	2	0
9	1	2	0	11	2	0	0	0	0	0

Table: NMF clustering

Pred/Ref	0	1	2	3	4	5	6	7	8	9
0	0	0	0	0	0	5	0	2	1	58
1	0	0	2	1	0	0	1	0	44	0
2	4	1	28	4	16	0	28	0	7	0
3	5	6	1	52	12	0	6	0	3	0
4	0	0	24	0	44	0	12	0	1	0
5	46	0	0	1	0	0	6	0	0	0
6	19	3	12	21	8	16	20	0	6	0
7	0	0	0	0	0	54	0	67	3	14
8	1	0	25	1	8	0	9	0	6	0
9	1	72	0	13	0	0	0	0	0	0

Table: k-mean clustering

- Tables show how the original labels are distributed amongst clusters for NMF and K-means
- Columns are the original labels (Ref) and rows (Pred) are the predicted clusters.
- For example, from the seventh reference label in the original data, the NMF clustering places 60 observations in cluster 0 and 9 observations in cluster 1. This is an example of a "good" cluster since most labels ended up in the same cluster.
- In terms of comparing NMF and k-means. Both methods perform well for some labels and worse for other labels. If you look at the tables together with the previous images and the orignal image, you can see that NMF seems to be better at clustering labels with low variance and kmeans is better at clustering labels with high variance.

Discussion

- Looking at the results, both methods do a fairly good job of clustering and the difference between the two methods are small.
- Looking at the cluster centroid images. The centroids for both methods are quite good at meaningfully represting their clusters. Taking into account that we only used a subset of the original data, the resulting images would probably have been much better had we used all the data.
- The kmeans clustering is slightly better at accurately representing clusters. However, this could be due to the fact that I only ran the NMF algorithm 10 times due to time constraints. The kmeans algorithm is much faster and I ran it with 25 different random starts (which is also quite low).
- The NMF vignette suggests that you should run the NMF algorithm 100-200 times but that would take too much time. Maybe this would have yielded better results?

Discussion

- I can't really explain why it seems like k-means is better at clustering labels with high variance, maybe its a coincidence. Or perhaps it could be due to the fact that k-means objective is to minimize **total within cluster variation** using the euclidian norm. NMF however, since it uses the Frobenius norm to minimize the objective function $X - HW$, the goal is basically to just **minimize everything**. In other words, minimize each entry (pixel) in X with its respective counterpart in HW. So I guess what im saying is that the goal of k-means is more aligned with clustering a label with high variance.
- Lastly, it is not surprising that we see similar results for both methods since the NMF clustering algorithm we are using (hard clustering) is basically the same as k-means clustering or at least a similar version of it. The W matrix from NMF is our centroids and the H matrix contains information of cluster membership.