Assignment of Foundation of Machine Learning (MSC)

Han Zhang hz5g21@soton.ac.uk

1 Non-negative matrix factorization

1.1 Introduction

Non-negative matrix factorization is used to decompose the non-negative large matrix into two non-negative small matrices, W and H. That means $V \approx WH$. The dimensions of V is $n \times m$, and the dimensions of the matrix factors W and H are $n \times r$ and $r \times m$, respectively. The r columns of W are called the base. Each column of H is called encoding and is in one-to-one correspondence with a face in V [2]. Our objective is to minimize the cost function which lists in Equation 1.

$$E(W, H) = \|V - WH\|_2 \tag{1}$$

The multiplicative update algorithm can be adopted in Equations 2 and 3 to minimize the cost function, where X represents the data that is required to factorize.

$$H = H \odot \frac{W^T X}{W^T W H} \tag{2}$$

$$W = W \odot \frac{W^T X}{W H H^T} \tag{3}$$

1.2 Factorization of synthetic data

In [2], The rank r of the factorization is usually selected by the rule $(n+m)r < nm$. So in this question, the r should be between 1 and 8. So r is setted to be 4,6 and 9 to test whether the algorithm fails. The results are shown in Figure 1.

(a) Rank 4 (b) Rank 6 (c) Rank 9

Figure 1: Convergence of the NMF algorithm

In choices of r being 4 and 6, the algorithm is always able to converge. However, in figure 1. c, it fails to reach convergence. This is because the initial choice of the matrix of W and H affect

the iteration count. If the max iteration is 80000, It will still convergence. In figure 1. a, It converges very quickly. it's probable that a small number of bases won't be able to reflect all of the data's information.

1.3 Comparison results with Sklearn

Compared to the results with the Sklearn, the error of own algorithm is $4.96e^{-15}$ which is almost the same with $6.52e^{-15}$ of Sklearn. The difference of factorization results W and H between own algorithm and the Sklearn is 7.12 and 1.87 by Frobenius Norm. Overall, those two algorithms have similar results.

1.4 Factor trading of FTSE 100

The W, $T \times r$, is determined after factorization of 95 component assets in the train dataset. The correlation coefficient between the components and the FTSE 100 is calculated using Pearson correlation. Table 1 shows the results.

Coefficient	**F1**	F2	F3	F4	F5	F6	F7	F8	F9	F10
FTSE 100	0.87	-0.77	0.61	0.77	0.34	0.72	-0.08	0.21	0.60	-0.55

Table 1: Coefficient between factors and FTSE 100

As Table 1 shows, column 1 is the most correlated one. To find out which stocks do contribute to this factor, the lasso linear regression is used to solve this problem. This is because that Using 95 constituent assets to predict the factor could figure out which stocks do contribution to this factor by sparse regularizer since it can do the feature selection. The λ of sparse regularizer is very important as it can decide how many stocks are left in linear regression and the accurate rate of predicting the factor since the bigger λ is, the more information lost. Besides duo to the transaction costs, 10 assets will be suitable. After the trial and error, The λ is set in 80. Figure 2 shows the results.

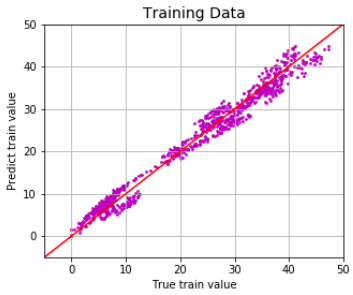


Figure 2: Linear regression to predict most correlated factors

The weights of stocks are shown in Table 2. All the stocks do contribution to the factor.

Stocks columns	0	4	5	22	28	37	41	57	65	91
Weight $\times$ 100	-0.01	0.84	0.75	-1.10	-0.48	-0.20	-0.17	1.07	0.33	-0.18

Table 2: Weight of stocks

For the evaluation of return, the investor is passive. The cumulative returns should be in equation 4.

$$Return = \frac{P_c - P_o}{P_o} \tag{4}$$

The cumulative returns of the FTSE100 index are 0.018, and the cumulative returns of the selected factor are -0.10. The selected factor has fewer cumulative returns than the FTSE100 index. This means that the selected stocks have a general inverse trend with FTSE100. The selected factors can represent the FTSE100 in the training dataset, but can not represent in the test dataset. It may be that lots of information are lost and the constituent companies in the FTSE 100 are not static. Some companies leave and new companies join in. Besides, the policies and other factors affect the results. Also, the lasso regularizer filters lots of information, the results may be not accurate.

For the performance, it is measured by the correlation coefficient between the selected price and the FTSE100 index since they have a better correlation, the selected stocks can be more representative. The coefficient between factor and FTSE100 index is -0.43 while The coefficient between an equally weighted basket of r and the FTSE100 index is 0.83. The random choice outperformed the selected factors.

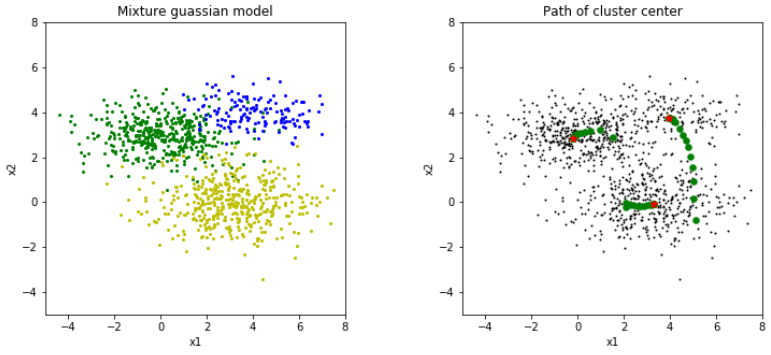
2 K-means

2.1 Introduction

The clustering algorithm K-means is widely used. It argues that the closer two targets are to each other, the more similar they are. Given the dataset $X = [X_1, X_2....X_n]$, The membership indicators are assigned to each data point. The cost function is demonstrated in equation 5.

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \|x_n - \mu_k\|_2 \tag{5}$$

2.2 K-means clustering



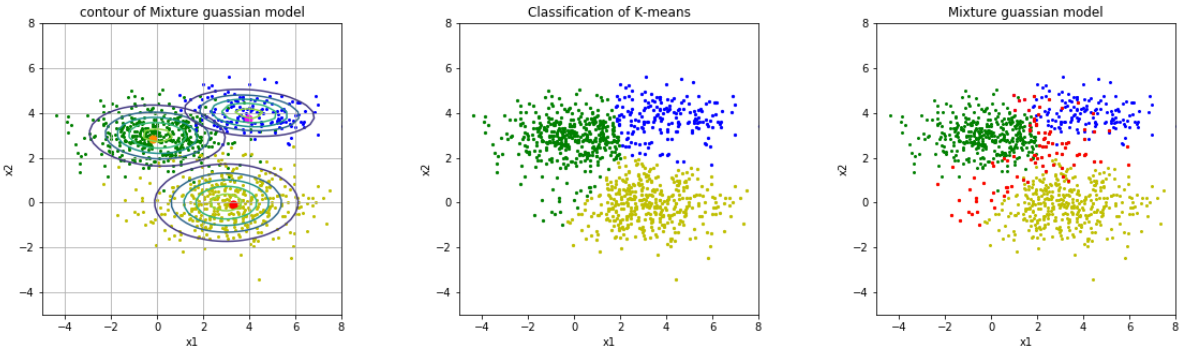
(a) Mixture Gaussians model (b) Convergence path

Figure 3: K-means implementation of Mixture gaussians model

In figure 3.a, the data from a mixture Gaussian density model are represented. The results of k-means are demonstrated in figure 3.b. The red points are the cluster centers from K-means

clustering. The green points show the initial guess of cluster centers and their convergence path during iterations.

2.3 Draw contours and comparison of regions



(a) contours of mixture model (b) Labeling after K-means (c) Wrong labeling

Figure 4: Boundary comparison

In figure 4, the original contours of probability density are shown in figure 4.a. It can be observed that there is some overlapping area. The data points in this area will lead to the wrong classification when using the K-means. Figure 4. b shows the data points cluster after k-means and figure 4.3 shows the wrong classification data points. It can be seen that the data points in the boundary are very easy to be the wrong classified. 10%,14%, and 2% of data belonging to each cluster are wrong classified.

2.4 Compare with Sklearn

Three measurement indicators are adapted to reflect the performance of the sklearn implementation and my method in order to compare them. The distance sum is utilised. It works out the sum of the squares of the distances between each data point and the nearest centre. The lesser the total, the better the performance. It is employed silhouette analysis, which is a way of interpreting and validating cluster consistency. The Silhouette coefficient ranges from -1 to 1, with a higher value indicating greater performance. Finally, the Calinski-Harabasz index approach is applied, which is based on dense and well-separated clusters. It's a lot faster than silhouette analysis, and a higher score equals better results.

	sun of Distance		Silhouette		Calinski Harabasz	
Experiment	Own method	Sklearn	Own method	Sklearn	Own method	Sklearn
1	2653	2653	0.4947	0.4948	1147	1148
2	2658	2658	0.5068	0.5068	1172	1172
3	2109	2109	0.5437	0.5437	1367	1367

Table 3: Performance comparison

The results show in table 3. It can be seen that the performance between my method and sklearn are nearly the same.

2.5 Initial center

Initial centroids have a significant influence on K-Means. Figure 5 shows how a bad selection of centroids might cause the algorithm to deliver inaccurate results.

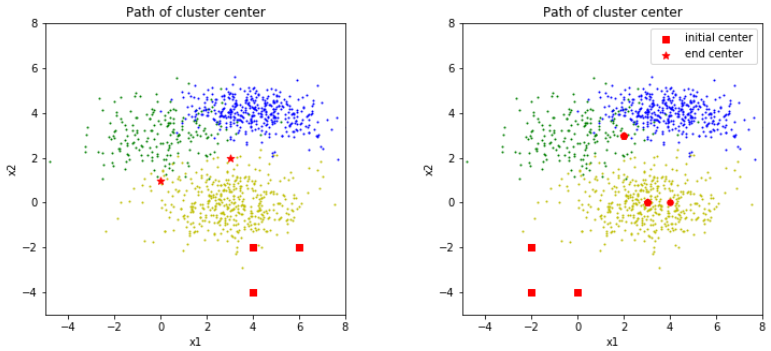


Figure 5: Bad selection of centroids

The Expectation-Maximization method is the key to K-Means. The bad selection is trapped at the local minimum, which is the cause of this problem. In Figure 5, the initial selection of centers is assigned to the bottom corner. Therefore, when K-Means re-average the centroids to minimize the total distance-to-centroid and maximize the likelihood, the centers are more likely to move to the nearest cluster. All the centroids are placed at the bottom, causing bad performance.

2.6 Selection of K

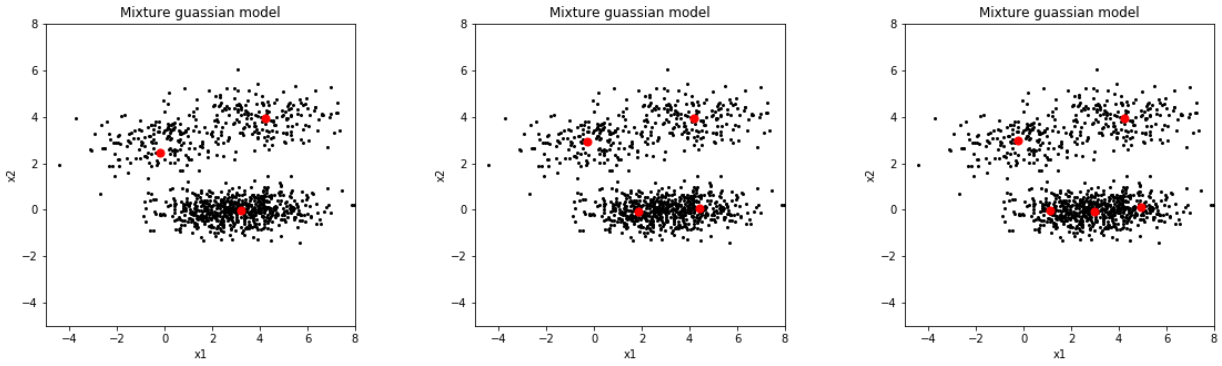


Figure 6: Selection of K

Multi-clusters in K-means are shown in Figure 6. As the value of K grows larger, the size of the cluster grows smaller, and the distance between clusters shrinks. However, a value of K that is too high may cause the cluster to be too closed. When K = 5, for example, three centroids are put in the bottom cluster.

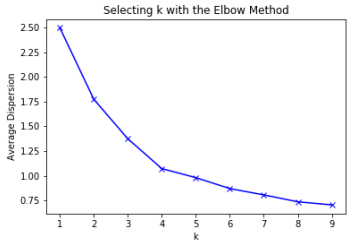


Figure 7: Elbow Curve of best K

One well-known strategy for selecting an appropriate K is to utilise the Elbow Curve[1]. As shown in Figure 7. K = 3 is the elbow of the curve, indicating the ideal value of K, according to the elbow curve.

2.7 Testing with Iris dataset

The Iris dataset are used to test our K-Means implementation in this part. The dataset is divided into three classes and it has four characteristics. The distribution shows in figure 8.

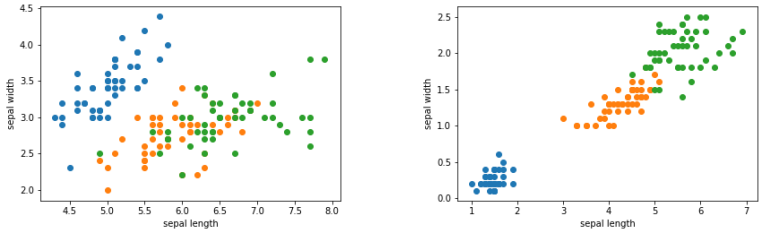


Figure 8: Iris data distribution

The cluster centroids shown in Figure 9 are obtained by clustering the dataset using K-Means (with K = 3). The algorithm's classification accuracy is 89.33%. It is a relatively good results.

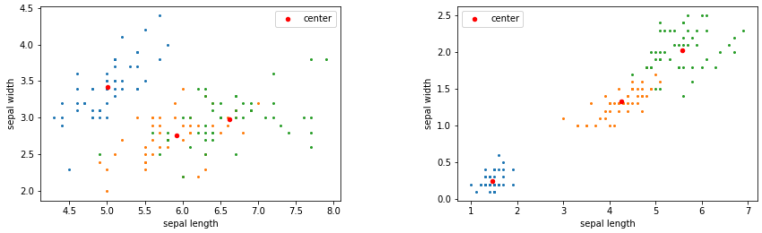


Figure 9: Centers in Iris dataset

References

[1] Purnima Bholowalia and Arvind Kumar. Ebk-means: A clustering technique based on elbow method and k-means in wsn. *International Journal of Computer Applications*, 105(9), 2014.

[2] Daniel D Lee and H Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.