1 Lab 1 - 5

I confirm I have completed all five labs and uploaded reports already. I confirm I didn't have 'Incomplete' that have been noted in the feedback of Lab1-2.

2 Non-Negative Matrix Factorization

In the beginning, it needs to complete the code and verified that the factorization is correct. To completed non negative matrix factorization code from the appendix is inserted the update W and update W code into it. From this code is updated W and H by used Multiplicative update rule Theorem 1 from [1] that match with Euclidean distance $||V - WH||$ function in the code.

$H_{\alpha\mu} \leftarrow H_{\alpha\mu} \frac{(W^T V)_{\alpha\mu}}{(W^T WH)_{\alpha\mu}}$, For update H

$W_{i\alpha} \leftarrow W_{i\alpha} \frac{(VH^T)_{i\alpha}}{(WHH^T)_{i\alpha}}$, For update W

The code will be completed when inserting this two-equation and now it can be verified. The first implemented that used rank r=6 and iteration is 40000, the convergence result of this NMF algorithm is illustrate as in Fig. 1.

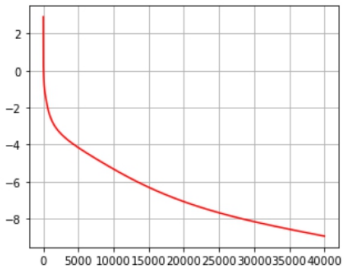

Figure 1: Convergence of the NMF algorithm (r=6)

Moreover, Fig. 2 and Fig. 3 present the implementation of NMF algorithm in rank r=3 and r=9, its show the minimize of Euclidean distance $||V - WH||$ of r=9 in convergence of NMF algorithm graph less than r=6 and r=3 is more than r=6. However, it can't show when minimizing Euclidean distance decrease.

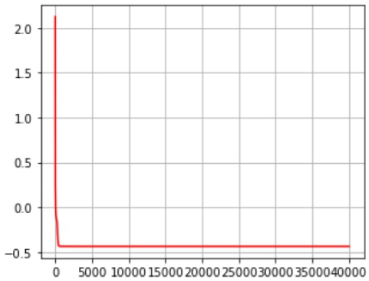

Figure 2: Convergence of the NMF algorithm (r=3)

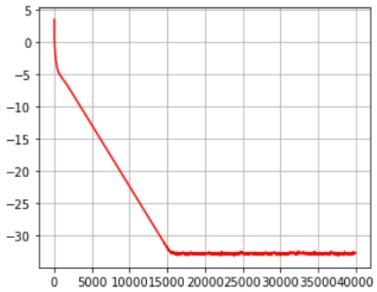

Figure 3: Convergence of the NMF algorithm (r=9)

To observe how to minimize Euclidean distance change in each rank I plot logarithm minimize Euclidean distance in rank 1-9 as present in Fig. 4 and plot of minimizing Euclidean distance in rank 1-9 as shown in Fig. 5. In addition, rank 1-9 is chosen because the rank not more than the length of the matrix column and row, for this implementation matrix column and row are 10 and 40, respectively.

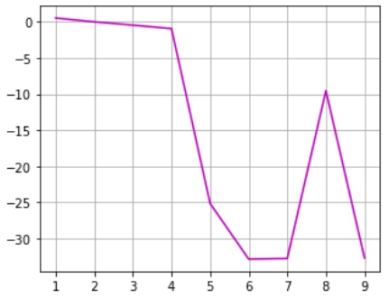

Figure 4: Plot of logarithm minimize Euclidean distance in rank 1-9 by NMF algorithm

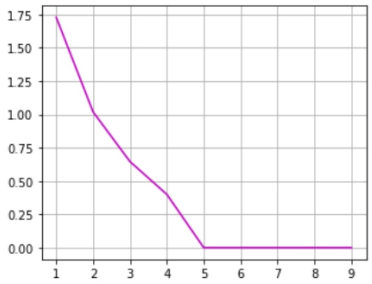

Figure 5: Plot of minimizing Euclidean distance in rank 1-9 by NMF algorithm

Fig. 6 and Fig. 7 is plot logarithm minimize Euclidean distance in rank 1-9 and plot of minimizing Euclidean distance in rank 1-9 by used nmf package in sklearn.decomposition [2]. When observing Fig. 5 and Fig. 7, it looks the same. when compare Fig. 4 and Fig. 6, it shows sklearn.decomposition can implement the result better than NMF algorithm since Fig. 4 was have

peak value in rank r=8 by random sometimes it appears. The peak value in rank r=8 might the algorithm fails.

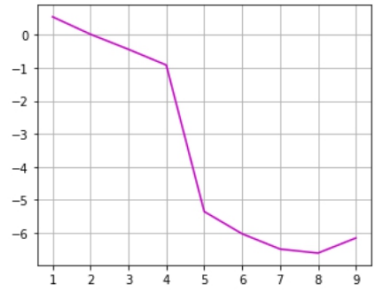

Figure 6: Plot of logarithm minimize Euclidean distance in rank 1-9 by sklearn.decomposition

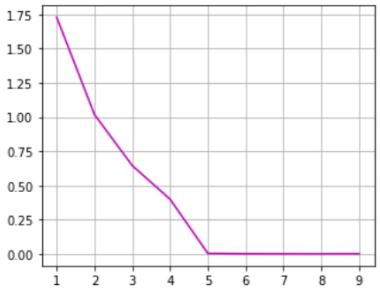

Figure 7: Plot of minimizing Euclidean distance in rank 1-9 by sklearn.decomposition

When the code is already completed and verified, the First is used this code to implement the daily FTSE100 stock index value dataset by separated into the first half is training data set and the second half is the testing data set. Next, the daily FTSE100 index data set will be plot as shown in Fig. 8 to checked and observed this data set.

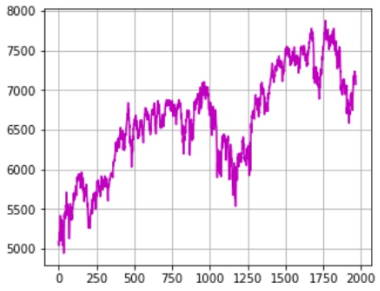

Figure 8: Plot of FTSE100 index during the period August 2011 to February 2019

After that, the daily FTSE100 index data set will be implemented by used the NMF algorithm with rank r=10 and iteration are 15000. The training data set and testing data set present in Fig. 9 and Fig. 10, respectively.

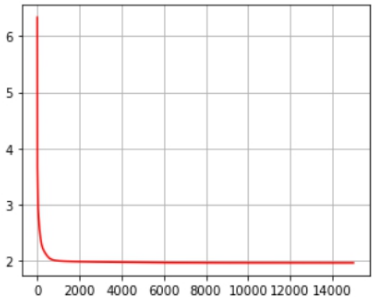

Figure 9: Convergence of the NMF algorithm with FTSE100 training dataset (r=10)

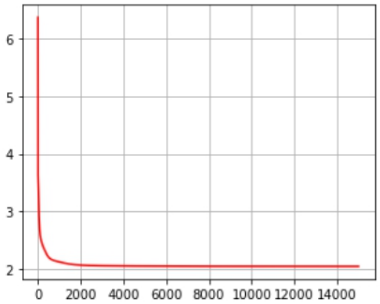

Figure 10: Convergence of the NMF algorithm with FTSE100 testing dataset (r=10)

Another result of implement testing data set is the $T \times r$ matrix from $T \times N$ data matrix that T is the number of time points, N is the number of assets and r is rank. The r columns in $T \times r$ matrix is cumulative and that normalizes to find a plot of correlation with normalizing FTSE100 index value as illustrate in Fig. 11. The correlation coefficient of Fig. 11 is 0.8964542346373656.

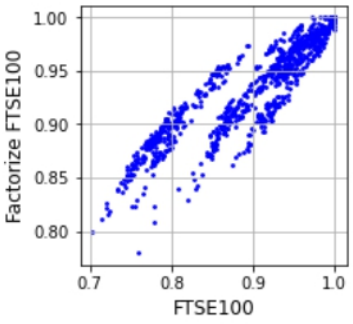

Figure 11: Plot of correlation between FTSE100 and NMF FTSE100 (r=10)

In Fig. 12, is the present plot of correlation between normalize FTSE100 index value and normalize equally weight basket of the same number with r. The correlation coefficient of Fig. 12 is 0.969845077546817.

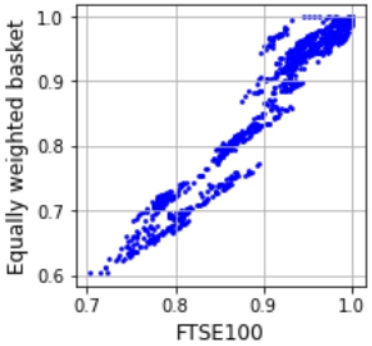

Figure 12: Plot of correlation between FTSE100 and equally weighted basket FTSE100 (r=10)

From Fig. 11, Fig. 12 and its correlation coefficient show equally weight basket of FTSE100 looked fit to FTSE100 than Non-Negative Matrix Factorization method that in this report called "NMF FTSE100". In Fig. 13 present plot of FTSE100, NMF FTSE100 and an equally weighted basket of FTSE100 that show NMF FTSE100 and equally weighted basket FTSE100 was the same trend with FTSE100 but NMF FTSE100 have trend line higher than FTSE100 and equally weighted basket FTSE100 have trend line lower than FTSE100. However, NMF FTSE100 can have more fit to FTSE100 by changing rank to find the better performance of NMF FTSE100 to fit with FTSE100.

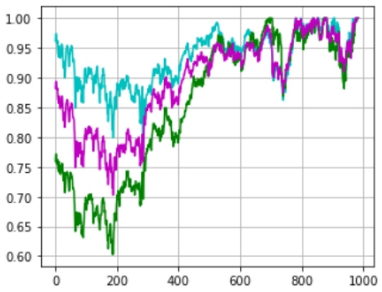

Figure 13: Plot of FTSE100, NMF FTSE100 and equally weighted basket FTSE100 (r=10)

The selected rank that fits FTSE100 and correlates most is rank $r = 30$ because it has a high correlation coefficient and not too much rank because the objective is actively selecting a small number of stocks to invest in. Next, compare the weight basket of FTSE100 and NMF FTSE100 again by observing in Fig. 14 and Fig. 15 its plot of correlation look seem similar.

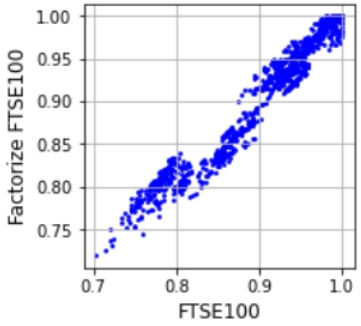

Figure 14: Plot of correlation between FTSE100 and NMF FTSE100 (r=30)

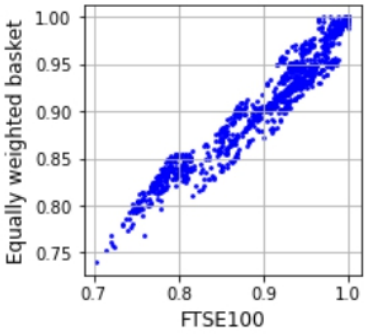

Figure 15: Plot of correlation between FTSE100 and equally weighted basket FTSE100 (r=30)

Consequently, it should compare by the used correlation coefficient that NMF FTSE100 has a correlation coefficient is 0.9867762059869185 that better than weight basket of FTSE100 have correlation coefficient is 0.9762363003444361. When compare the plot of FTSE100, NMF FTSE100 and the equally weighted basket of FTSE100 as illustrate in Fig. 16, all plots all in the same trend. However, when implementing the code much time sometimes gets Fig. 17 that equally weighted basket of FTSE100 not similar to the FTSE100 index.

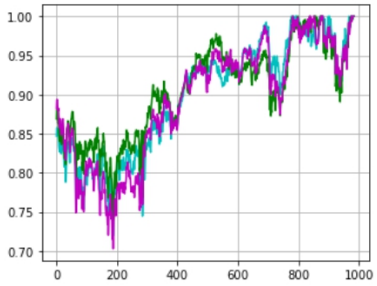

Figure 16: Plot of FTSE100, NMF FTSE100 and equally weighted basket FTSE100 (r=30)

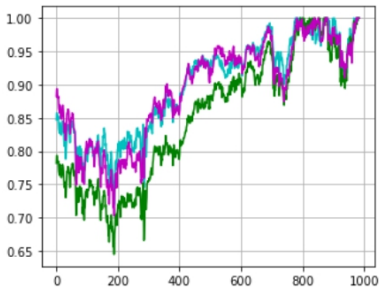

Figure 17: Plot of FTSE100, NMF FTSE100 and equally weighted basket FTSE100 (r=30)

Conclusion

Non-Negative Matrix Factorization is very useful to reduce the dimension of the data set and applied it to use in many ways. NMF algorithm just has the same problem to enhance when compare to nmf package in sklearn.decomposition. For the FTSE100 index, the NMF algorithm is suitable to reduce the number of stocks to invest in than an equally weighted basket of stock in FTSE100.

Reference

[1]: D. Lee and H. S. Seung, Algorithms for Non-negative Matrix Factorization. NIPS'00: Proceedings of the 13th International Conference on Neural Information Processing Systems, 2000. [Online]. Available: `http://web.cs.ucla.edu/~yzsun/classes/2014Spring_CS7280/Papers/Clustering/NNF_lee01algorithms.pdf` [Accessed December, 03 2020].

[2]: Scikit-learn.org. 2020. Sklearn.Decomposition.NMF — Scikit-Learn 0.23.2 Documentation. [online] Available at: `https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.NMF.html` [Accessed December, 03 2020].