Artificial Intelligence in Fintech Project (1): [1]

[1] Feel free to do it in a group or individual way

Introduction

Dimension reduction has not be seriously investigated in finance area though recent some efforts were made. PCA is a widely used one in finance area for visualization, but full interpreting of PCs are almost not found in finance. Furthermore, other dimension reduction methods are rarely found in the literature (e.g., t-SNE, KPCA, SPCA)

In this project, you are going to explore the following dimension methods besides PCA for finance data or data in related finance areas. I believe it will be the most comprehensive dimension reduction method investigation in finance. Except SPCA and PCA, all methods belong to manifold learning methods.

- t-SNE: t-SNE minimizes the Kullback-Leibler divergence between two distributions P and Q that measure two pairwise similarities in different spaces. Unlike PCA, t-SNE is reported to be more robust with respect to the presence of outliers

- UMAP: has a more robust theoretical foundation rooted in Riemannian manifold and tangent space theory, but it can be viewed as an optimized version of t-SNE. UMAP assumes data is uniformly distributed on a Riemannian manifold. UMAP employs different techniques to compute data similarity in the high-dimensional input space and low-dimensional embedding.

- KPCA: It is the nonlinear extension of PCA in a high-dimensional space via kernel tricks. It will bring more features for finance data along with some interpretation issue in visualization. It has a high requirement for data clean degree.

- SPCA: It imposes more zeros on each PCA to increase the interpretation of latent data characteristics representation in dimension reduction.

- LLE claims to be able to generate highly nonlinear embeddings for high-dimensional data while keeping the underlying data structure than PCA and MDS besides avoiding local minima issue in optimization. However, it can distort the whole global data geometrical structure for its nearest neighbor searching mechanism.

- ISOMAP aims to keep the geodesic distances between points under dimensional reduction. The geodesic distance is the shortest path between two points on a curved surface. Compared with LLE, it can keep the whole global data structure, but it suffers from high complexities and can be sensitive to noise data

A) HFT and cryptocurrency trading marker discovery (100 points)

The trading marker is a meaningful buying or selling point for profitability in trading. HFT can do trading many times frequently for a marginal price.

We say price $p(t)$ is a trading marker at time t if its change rate with respect to is $\geq$ a threshold $\eta(0.5\%)$ in a interval $[t_1, t_2]$, say, $|p(t) - p(t')|/|p(t')| \geq 0.5\%$, $t, t' \in [t_1, t_2]$. The simplest case will be $t' = t - 1$: the change rate in an time interval $\geq 0.5\%$.

- It is up to you to set threshold to determine your trading marker threshold, say 0.5%.

1. Apply the trading marker discovery method (M-SCAN) I provided in the lecture note to find the markers of HFT and cryptocurrency datasets given in this project.
2. You need to visualize your datasets in a representative way.
3. Feel free to do feature selection by yourself: you are encouraged to do feature interpolations, i.e., infer new features and add them into the datasets
4. You need to use at least four dimension reduction methods in your project but must include PCA, UMAP, and t-SNE.
5. Visualize the outliers by doing biplot and triplots of the dimension reduction algorithms
   (a) You need to show the outliers detected by the methods + DBSCAN
   (b) You need to tune the parameters to get better results
   (c) You can do feature interpolations.
6. You need to report how many outliers become 'true markers' in your implementations

(a) How to get the true markers?

i. Do FFT for your HFT/cryptocurrency data and sort the real parts of the FFT coefficients.

ii. The largest FFT coefficients will correspond to the true markers.

iii. Although I did not prove it theoretically, it works very well in practice.

iv. You need to draw the predicted markers in the original price curve to validate it.

7. You need to calculate the mean square error of your predicted markers between the true markers and your predicted markers in terms of their prices.

8. Draw your conclusion for the results that at least include

   (a) which dimension reduction methods work well in marker discovery

   (b) why different sector HFT data has so different performance in trading marker discovery

9. Why cryptocurrency data behave different from HFT data in trading marker discovery.

10. Use the codes I give in lecture or your codes/methods to retrieve at least two new HFT datasets and test your results.

B) Develop c-SNE methods for finance data (100 points)[2]

t-SNE evolves from the SNE method by using the student-t distribution to replace the original normal distribution in the embedding space.

Develop a c-SNE method by replacing both normal distributions and t-distributions in the original t-SNE to get a c-SNE method for finance data. You can refer the original python codes for t-SNE

Cauchy distributions have high peaks and long tails compared to the traditional normal distributions. Such characteristics are good for modeling stock data in finance.

Compare c-SNE with t-SNE on four datasets at least two of them should be stock data and draw your conclusion

[2]Extra credits

What should you turn in?

- 1. A folder that contains
  - A ppt to show details of your analytics (at least 50 pages)
  - your data
  - source files
  - corresponding related output.
- 2. Send the zipped file (.zip instead of ,rar) of your folder to Canvas before the DUE