TCIO GLOBAL PORTFOLIO STRATEGY PROJECT

DONGXU LI

Part 1) Query and retrieve and process the data from FRED.

Treasury Constant Maturity Rate data is retrieved through python API: fredaip. The process is fairly straight forward, the processed data is in pandas dataframe, the data is clean and has no missing values. The data contains all rate values for 6 tickers from a time window through 1989-01-01 to 2019-10-01, the data frequency is monthly. The head and tail of the dataframe is shown below:

Sample Data

Date	GS1	GS2	GS3	GS5	GS7	GS10
1989-01-01	9.05	9.18	9.20	9.15	9.14	9.09
1989-02-01	9.25	9.37	9.32	9.27	9.23	9.17
...	...	...	...	...	...	...
2019-08-01	1.77	1.57	1.51	1.49	1.55	1.63
2019-09-01	1.80	1.65	1.59	1.57	1.64	1.70

Summary statistics of the time series data is shown below:

Summary Statistics

	GS1	GS2	GS3	GS5	GS7	GS10
count	370.00	370.00	370.00	370.00	370.00	370.00
mean	3.22	3.52	3.74	4.12	4.42	4.65
std	2.48	2.48	2.41	2.26	2.15	2.02
min	0.10	0.21	0.33	0.62	0.98	1.50
25%	0.65	1.03	1.48	1.98	2.42	2.82
50%	3.21	3.49	3.76	4.15	4.34	4.53
75%	5.33	5.58	5.76	5.93	6.15	6.16
max	9.57	9.68	9.61	9.51	9.43	9.36

Part 2) Description of the dimension reduction technique.

Before I begin to describe the dimension reduction technique, I make the following assumptions. From the original dataframe, I compute a new dataframe denoting the rate of change in the value of each ticker. Then assume that the rate of change of each time series data is a random variable with a fixed distribution, for example, if we denote the value of GS1 at time t be $V_t(GS1)$ then we have:

$$\frac{V_t(GS_1) - V_{t-1}(GS_n)}{V_{t-1}(GS1)} \sim x_1$$

.

Where x_1 is the random variable, we assume x_1 follows a fixed distribution within a relatively short time window. Then we model the rate of change for every ticker $i \in \{1, 2, 3, 5, 7, 10\}$:

$$\frac{V_t(GS_i) - V_{t-1}(GS_i)}{V_{t-1}(GS_i)} \sim x_i$$

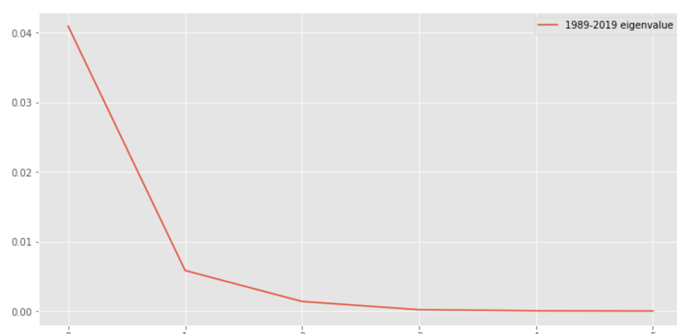
.
Where x_i is also a random variable, we assume each x_i has a different but also fixed distribution. From the random variables, we can construct a random vector $X = (x_1, x_2, x_3, x_5, x_7, x_{10})^T$, then each data point X_t where $t \in [1, T]$ in the dataframe is just an independent copies (or realization) of X.

With this setup, we could apply Principal Componet Analysis Algorithm to project the high dimensional data into a low dimensinal linear subspace which still explains "most" of the variation in it :

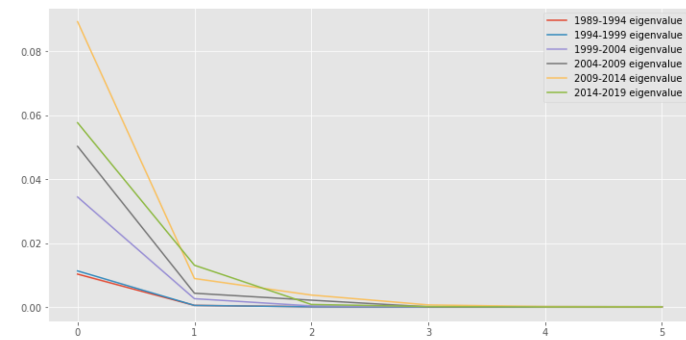
(1) Input: $X_1, ..., X_T \in R^d$: cloud of T points in dimention $d = 6$.
(2) Step1: Compute the empirical covariance matrix.
(3) Step2: Compute the eigenvalue decomposition $S = PDP^T$, where $D = Diag(\lambda_1, ..., \lambda_6)$, with $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_6$ and $P = (v_1, v_2, ..., v_6)$ is an orthogonal matrix where each column is the eigen vector corresponding to each eigen value.
(4) Step3: Choose $k < d$ and set $P_k = (v_1, v_2, ..., v_k)$.
(5) Output: $Y_1, ..., Y_T$ where $Y_t = P_k^T X_t \in R^k$, and $t = 1, ..., T$.

Following the PCA algorithm: 1) I input the rate of change data for tickers (the entire dataset). 2) Compute the empirical covariance matrix. 3) Conduct the eigenvalue decompositoin. And then 4) I take a look at the eigen values of the sample covariance matrix (scree plot):

The vertical axis mesures eigenvalues, the horizontal axis denotes the descending orders of 6 eigenvalues.



From the plot above, I pick the value of $k = 2$, since I intuitively view λ_2 as an inflection point in the sequence $\lambda_1, \lambda_2, ..., \lambda_6$ and I believe a linear subspace spanned by v_1 and v_2 will capture most of the variance in the data points. If we ran this eigenvalue decomposition on covariance matrices calculated from different time periods, it seems that the selection of value $k = 2$ will not change. Then the selection of reduced dimensions are not sensitive to the change of time. To provide more evidence, I show the scree plot of eigenvalues during different time periods:



How do I interpret these "reduced" dimensions? This reduced dimension captured the most part of variation in the original treasury data, gives us a view over the entire treasury data which is easier to visualize and analyze.

Part 3) method used for the similarity between data series.

For similarity between data series, measure the timeseries difference by Euclidean is an obvious choice, but here I tried a difference approach using Dynamic Time Warping, the detail of the algorithm is described in Berndt's paper. Here I implimented the algorithm and calculated average DTW between 6 time series. It is a rough measure of the difference between time series.

$$mean\text{DTW}=\frac{\Sigma_{i,j}\, DTW(S_i, S_j)}{C_6^2}$$

.

Then I calculate the mean DTW value for every year and get the following results:

meanDTW	year
0.369	1989
0.964	1990
...	...
0.858	2018
0.328	2019

min value of meanDTW	year
0.328	2019

The smallest value of average DTW is found in 2019, so I conclude that in the year of 2019 it is most similar – in terms of the 6 data series.

Reference

Berndt, D.J. and Clifford, J., 1994, July. Using dynamic time warping to find patterns in time series. In KDD workshop (Vol. 10, No. 16, pp. 359-370).

Wold, S., Esbensen, K. and Geladi, P., 1987. Principal component analysis. Chemometrics and intelligent laboratory systems, 2(1-3), pp.37-52.