Báo cáo bài tập thực hành Xử lý đa chiều 01

Nguyễn Phú Thành - MSSV: 18110014

17/5/2021

- **Đọc và xử lý dữ liệu có chứa time series**

Bộ dữ liệu room-temperature.csv gồm 4 cột: FrontLeft, FrontRight, BackLeft, BackRight chỉ nhiệt độ của 4 góc của một căn phòng, thời điểm đo nhiệt độ được ghi lại ở cột Date.
Để parse cột thời gian, ta sử dụng thư viện datetime với hàm datetime.datetime.strptime cùng với thư viện pandas:

```
    pd.read_csv(
        'Ten file',
        parse_dates = True,
        date_parser = ...,
        index_col = 'Date'
    )
```

Trong đó, tham số date_parser là một hàm dùng để chuyển một chuỗi thành dạng time series
Nhận thấy một trong các giá trị của cột Date có dạng '4/11/2010 11:30', tức có dạng 'Tháng/Ngày/Năm Giờ:Phút'. Tra bảng chuyển đổi trong documentation của datetime (Link documentation), chuỗi 'Tháng/Ngày/Năm Giờ:Phút' được chuyển thành chuỗi '%m/%d/%Y %H:%M'
Khi đó ta truyền tham số date_parser như sau:

```
    date_parser = lambda s: datetime.datetime.strptime(s, '%m/%d/%Y %H:%M')
```

Năm dòng đầu của DataFrame sau khi đọc dữ liệu như trên:

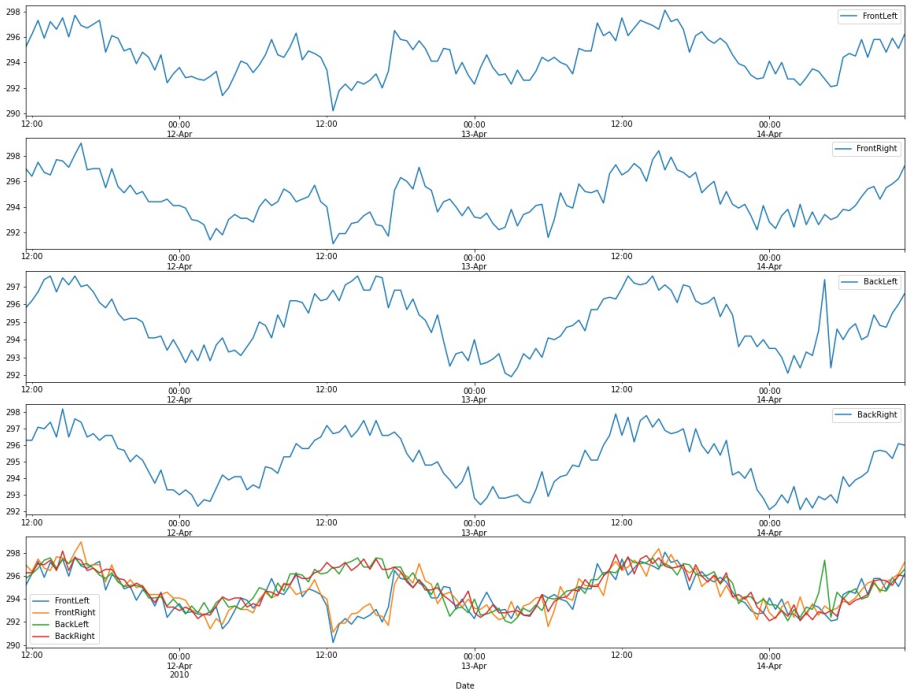
```
In [3]: data.head()
```

Out[3]:

Date	FrontLeft	FrontRight	BackLeft	BackRight
2010-04-11 11:30:00	295.2	297.0	295.8	296.3
2010-04-11 12:00:00	296.2	296.4	296.2	296.3
2010-04-11 12:30:00	297.3	297.5	296.7	297.1
2010-04-11 13:00:00	295.9	296.7	297.4	297.0
2010-04-11 13:30:00	297.2	296.5	297.6	297.4

- **Biểu đồ Time Series**

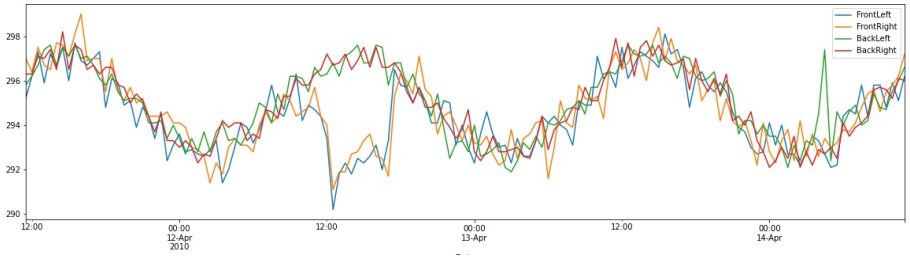
Ta vẽ biểu đồ đường thể hiện nhiệt độ của từng góc phòng theo thời gian cùng với biểu đồ đường thể hiện nhiệt độ của 4 góc phòng trong cùng 1 plot



Hình 1: Biểu đồ Time Series biểu thị nhiệt độ 4 góc phòng theo thời gian

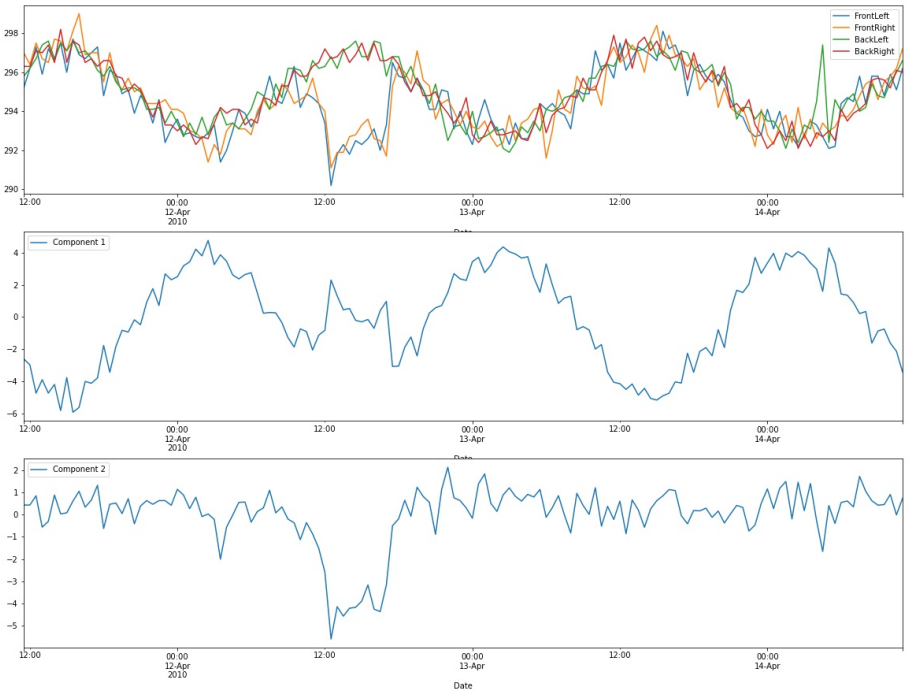
- **Nhận xét biểu đồ:**

Nhận thấy biểu đồ đường nhiệt độ (đặc biệt ở biểu đồ đường nhiệt độ của cả 4 góc phòng phòng trong cùng một plot) chủ yếu có **dạng đường hình *sin*** cùng với **một phần nhiễu từ các phép đo nhiệt độ**



Hình 2: Biểu đồ đường nhiệt độ của 4 góc phòng trong cùng 1 plot

Như vậy, ta sẽ chiếu dữ liệu xuống **không gian 2 chiều** trong thuật toán PCA mà vẫn giữ được những thông tin quan trọng của dữ liệu. Khi vẽ biểu đồ nhiệt độ cùng với biểu đồ các giá trị của hai thành phần chính, ta cũng thấy 2 xu hướng này



Hình 3: Biểu đồ nhiệt cùng với biểu đồ giá trị 2 thành phần chính

Thành phần chính thứ nhất có dạng hình sin trong khi thành phần chính thứ hai dao động nhiều hơn, thể hiện nhiễu của dữ liệu

• **Cài đặt PCA**

Ta cài đặt PCA qua các bước sau:

1. Cho ma trận $\boldsymbol{X}$ gồm n dòng, p cột. Ta tính vectơ trung bình $\bar{\boldsymbol{X}}$ có p chiều. Cho mỗi dòng của ma trận $\boldsymbol{X}$ trừ cho vectơ trung bình này được ma trận $\boldsymbol{X'}$ có trung bình là vectơ $\mathbf{0}$
2. Tính ma trận hiệp phương sai: $\boldsymbol{S} = \dfrac{1}{n-1}\boldsymbol{X'}^T\boldsymbol{X'}$
3. Tìm các cặp trị riêng, vectơ riêng $(\lambda_i, \boldsymbol{e}_i)$ của ma trận $\boldsymbol{S}$
4. Sắp xếp các cặp trị riêng, vectơ riêng này theo chiều giảm dần của trị riêng
5. Với k là số chiều mà ta muốn chiếu xuống, chọn k cặp trị riêng, vectơ riêng đầu tiên
6. Sắp ma trận chiếu $\boldsymbol{W}$ có số chiều là $k \times p$, trong đó mỗi dòng là một vectơ riêng trong k vectơ riêng đã chọn ở bước trên
7. Ma trận $\boldsymbol{Y} = \boldsymbol{X'}\boldsymbol{W}^T$ chính là biểu diễn của $\boldsymbol{X}$ qua phép chiếu $\boldsymbol{W}$ (sau khi trừ cho trung bình)

Bước 3 và bước 4 đã được thư viện numpy hỗ trợ. Với lưu ý rằng, vì $\boldsymbol{S}$ là ma trận đối xứng nên thay vì sử dụng trực tiếp hàm eig của numpy.linalg, ta sẽ sử dụng hàm eigh của numpy.linalg để

đảm bảo tính ổn định (numerical stable) (Link documentation của eigh)
Mặt khác, hàm eigh trả về các trị riêng, vectơ riêng theo chiều *tăng dần* mà ta lại cần chiếu *giảm dần* nên ta phải đảo chiều các giá trị trả về

```
    eigenvalues, eigenvecs = np.linalg.eigh(S)
    eigenvalues, eigenvecs = eigenvalues[::-1], eigenvecs[:, ::-1]
```