IPCA_main

May 11, 2019

```
In [1]: import altair as alt
        import pandas as pd
        import numpy as np
        import matplotlib.pyplot as plt
        import seaborn as sns
```

1 Read data

```
In [5]: df_ = pd.read_csv("../Data/Characteristics Freyberger et al.csv", index_col = 0)

In [13]: print("Column names:\n", list(df_))
         df = df_.iloc[np.random.choice(df_.shape[0], 100),:]
         print("Sub-sampled dataframe shape:", df.shape)
         df.head()
```

```
Column names:
 ['yy', 'mm', 'date', 'permno', 'ret', 'q10', 'q20', 'q50', 'prc', 'a2me', 'ato', 'beme', 'c',
Sub-sampled dataframe shape: (100, 45)
```

```
Out[13]:          yy  mm     date  permno       ret          q10          q20  \
        12209   1990   4  4/30/90   10198 -0.250000   44663.1000   99075.7500
        840519  2007   3  3/31/07   65285 -0.192555  369366.4632  746434.2442
        293795  1981   7  7/31/81   25697 -0.060420   41245.5000   74047.5000
        515861  1984   4  4/30/84   41575  0.095484   55173.5250   93946.6750
        274919  1982   8  8/31/82   24336  0.176320   33596.8125   55271.7000

                        q50      prc      a2me  ...      beta  cum_return_12_2  \
        12209    494892.125   -0.375  3.491766  ... -0.142502        -0.238095
        840519  2358321.476   25.160  2.020736  ...  1.898876        -0.278031
        293795   266185.875   33.500  0.714904  ...  1.160661         0.470630
        515861   333984.000  -33.750  0.478822  ...  0.594358        -0.182078
        274919   210015.000   18.000  3.185968  ...  0.743310        -0.108717

                cum_return_12_7  cum_return_1_0  cum_return_36_13  idio_vol  \
        12209          0.190476        0.000000         -0.700000  0.020537
        840519        -0.425504       -0.063983          0.142893  0.025153
```

```
293795          0.483350       -0.017182          0.378623  0.029692
515861         -0.075206       -0.120567          0.817200  0.005606
274919          0.023335        0.024590          0.214467  0.010364

        spread_mean       suv  rel_to_high_price       lev
12209      0.239394 -0.076449           0.500000  0.655746
840519     0.000735  1.875480           0.664392  0.497087
293795     0.024296 -0.263903           0.846154  0.360177
515861     0.016056 -0.107709           0.606357  0.091218
274919     0.015761 -0.518771           0.811688  0.463971

[5 rows x 45 columns]
```

In [11]:
```
print(df_.shape)
np.unique(df_['permno']).shape[0]
```

```
(1048575, 45)
```

Out[11]: 7593

2 Teminology

3 Findings (quotes)

- IPCA is a competitive model for describing the variability and hence riskiness of stock returns
- In summary, IPCA is the most successful model we analyze for jointly explaining realized variation in returns (i.e., systematic risks) and differences in average returns (i.e., risk compensation).
- By linking factor loadings to observable data, IPCA tremendously reduces the dimension of the parameter space compared to models with observable factors and even compared to standard PCA.
- To be continued

4 Model specification

$r_{i,t+1} = \alpha_{i,t} + \beta_{i,t} f_{t+1} + \epsilon_{i,t+1}$
$\alpha_{i,t} = z'_{i,t}\Gamma_\alpha + \nu_{\alpha,i,t}, \beta_{i,t} = z'_{i,t}\Gamma_\beta + \nu_{\beta,i,t}$
where f is factor vector; z is characteristics factor.

5 Implementation

5.1 Restricted model $\Gamma_\alpha = 0$

Given information up to time t, the return vector (size $N \times 1$) can be written as

$$r_{t+1} = Z_t \Gamma_\beta f_{t+1} + \epsilon^*_{t+1}$$

where r_{t+1} is the return vector; Z_t is $N \times L$ matrix of characteristics.

Alternating least squares: - If Γ_β is known. Need to solve $r_{t+1} = Z_t \Gamma_\beta f_{t+1}$. Use $\hat{f}_{t+1} = \left(\Gamma_\beta^T Z_t^T Z_t \Gamma_\beta\right)^{-1} \Gamma_\beta^T Z_t^T r_{t+1}$ - If f_t is known. Need to solve

```
In [ ]:
```