Motivation of Factor Models

Factors models are widely used in industry and serve two main purposes.

The first is to reduce the complexity of modeling asset price movements.

For instance, trying to build a model that completely explains stock price movements is near impossible.

In order to build a model for our favorite stock will need to model supply, demand, sentiment, current and expected future earnings of the stock, news, interest rates, risk premium.

It's near impossible to calibrate such a complicated model!

Instead, factor investors assume that there are N numbers of important factors that drive a portion of the asset returns.

They then say that at the portfolio level, asset specific movements can be averaged out, and only those N variables remain.

So to understand what drives the portfolio returns we only need to model the effect of that small number of factors.

Alternatively, understanding the factor loadings of the individual assets allows us to estimate the covariance of our returns.

We state without proof that if one understands the factor loadings and the covariance of the factor returns, one can then compute an estimate for the covariance of the assets themselves.

Secondly, factor models can also be used for hedging.

We again state without proof that the factor loadings represent the hedging ratio one would use to minimize the volatility of our portfolio.

Here, we will walk through multiple ways of estimating factor loadings, and discuss their relative strengths and weaknesses.

In this example, we are interested in explaining the asset returns with a five-factor model:

1) World Equity: This factor represents worldwide equity returns.

2) US Treasury: This factor contains return from treasury bonds in United States, the bonds with the least risk.

3) Bond Risk Premium: This is a credit factor that captures extra yield from risky bonds. Defined as the spread between high risk bonds and US Treasury bonds.

4) Inflation Protection: This is a "style" factor that considers the difference between real and nominal returns, thus balances the need for both.

5) Currency Protection: This is also a "style" factor that includes risk premium for US domestic assets.

Our Asset classes are: US Equities, Real Estate, Commodities and Corp Bonds

OLS Regression

Building a factor module is equivalent to solving for the 5 factor loadings defined above.

Ordinary Least Squares (OLS) regression is the simplest way.

OLS regression is equivalent to solving the following optimization problem:

$$\hat{\beta}^{OLS} = argmin_{\beta} \left\{ \sum_{i=1}^{n} (y_i - X_i * \beta)^2 \right\}$$

In our notation, n in the number of data points. In this case, OLS regression has a closed form solution.

$$\hat{\beta}^{OLS} = (\mathbf{X_t}^T \mathbf{X_t})^{-1} \mathbf{X_t}^T \mathbf{Y_t}$$

Where $\mathbf{Y_t}$ is the vector representation of y_t, and $\mathbf{X_t}$ is the matrix representation of X_t

β is the factor loadings…..They represent the effect on the dependent variable caused by movement in the underlying factor.

After Building OLS Regression Model, we arrive at the following output:

OUTPUT1:

```
Dependent variable is Real Estate
             Intercept  World Equities  US Treasuries  Bond Risk Premium  \
theory_beta  -0.001734        0.690712       1.338867           0.976594

             Inflation Protection  Currency Protection
theory_beta              0.217813             1.004761
```

Here, we can see that Real Estate is our Asset class and it was our Target variable .whereas the five factor loadings were independent variables.

.we can see that World Equity impacts Real Estate by 0.690712

Similarly US Treasury impacts Real Estate by 1.338867 and so on…..

****
****

Now we can build other OLS Regression Model using the Asset class

US Equities as our Target variable .whereas the five factor loadings as independent variables and get the following output:

OUTPUT2:

```
Dependent variable is US Equities
             Intercept  World Equities  US Treasuries  Bond Risk Premium  \
theory_beta   0.000049        0.536876       0.579876            0.61014

             Inflation Protection  Currency Protection
theory_beta              0.083233             0.625217
```

Similarly we can build OLS Regression Model using the other Asset classes namely Commodities and Corp Bonds

Kindly note that we get similar results if instead of Building Models using OLS we build our model using Linear Regression via Scikit-learn.
from sklearn.linear_model import LinearRegression

OLS Drawbacks

Firstly OLS has no mechanism to filter out noise variables. We will have many noise variables that need to be removed.

Secondly, it assumes that factor loadings are constant over time. We will know factor loadings are highly dependent on the time period.

DEMONSTRAING FIRST OLS DRAWBACK:

We will introduce a noise variable positively correlated with the World Equities factor. Then we have re-run the OLS regression.

The OLS regression chooses to average the two signals, changing the loading on the World Equity factor.

OUTPUT3:

After Building OLS Regression Model with noise variable positively correlated with the World Equities factor**, we arrive at the following output:**

```
Dependent Variable is Real Estate
Time period is between January 1985 to September 2018 inclusive
                Intercept  World Equities  US Treasuries  Bond Risk Premium  \
OLS with Noise  -0.003654        0.323065       1.214039           1.054216

                Inflation Protection  Currency Protection     Noise
OLS with Noise              0.283488             0.361987 -0.010582
```

COMPARING OUTPUT3 WITH OUTPUT1, WE CAN CLEARLY SEE THAT IMPACT OF World Equity Factor Loadings ON Asset class REAL ESTATE SIGNIFICANTLY changed after we intentionally added a noise factor to World Equities factor.

DEMONSTRAING SECOND OLS DRAWBACK:

We will filter our data into two different regimes.

The first regime "normal", will be months where US Equities had a positive monthly return.

The second regime, "crash" will be months where US Equities had a negative return.

We can see different factor loadings for Crash periods and Normal Periods

OUTPUT4:

After Building OLS Regression Model for Normal Regime, we arrive at the following output:

```
normalData = all_data[all_data['US Equities'] > 0].copy()
```

```
Dependent Variable is Real Estate
Time period is between January 1985 to September 2018 inclusive
            Intercept  World Equities  US Treasuries  Bond Risk Premium  \
OLS Normal    0.00348         0.15022        1.30286           1.100558

            Inflation Protection  Currency Protection
OLS Normal              0.210193              0.05183
```

OUTPUT5:

After Building OLS Regression Model for Crash Regime, we arrive at the following output:

```
crashData = all_data[all_data['US Equities'] <= 0].copy()
```

```
Dependent Variable is Real Estate
Time period is between March 1985 to June 2018 inclusive
           Intercept  World Equities  US Treasuries  Bond Risk Premium  \
OLS Crash  -0.011274        0.365824       0.959781           0.792532

           Inflation Protection  Currency Protection
OLS Crash              0.540801              0.588727
```

Alternative ML Methods

Now that we have seen the drawbacks of OLS, we can move to modern machine learning techniques. We will discuss new methods in this section as improvements to OLS to handle noise variables.

The first, LASSO regression, is the simplest version of regularized regression. Regularized regression means that we add a penalty term to the optimization problem to penalize the model's complexity.

For the sake of intuition, imagine if we could penalize the number of non-zero coefficients we add to the model. In that instance we would expect that the model would only consider the variables that really influence Y, and ignore the noise variables.

LASSO regression does exactly that.

LASSO REGRESSION

$$\hat{\beta}^{\text{LASSO}} = argmin_{\beta}\left\{\sum_{i=1}^{n}(y_i - X_i * \beta)^2 + \lambda \sum_{j=1}^{m}|\beta_j|\right\}$$

n is the number of data points, and m is the number factors.

it is the same as the OLS regression, but with a second penalty term. As stated above, it directly penalizes the use of non-zero coefficients and can be shown to set factor loadings to zero.

$$\frac{\lambda}{2 * n} = \alpha$$

Here λ is called a hyper parameter which we need to choose. Scikit-learn don't use λ it uses α which can be related via the equation as above.

Where n is the number of data points. For now, let's arbitrarily pick λ=.1

OUTPUT6:

After Building Lasso Regression Model without noise and λ=.1, we arrive at the following output:

```
Dependent Variable is Real Estate
Time period is between Febuary 2001 to September 2018 inclusive
lambda = 0.1
                Intercept  World Equities  US Treasuries  Bond Risk Premium  \
LASSO Regression  0.006142        0.806932       0.023147                0.0

                Inflation Protection  Currency Protection
LASSO Regression                 0.0                  0.0
```

Notice that LASSO sets factor loadings to zero. Because of this, it can filter out noise variables. Let's rerun the analysis including the noise variable.

Here LASSO sets factor loadings to zero. Because of this, it can filter out noise variables. Let's rerun the analysis including the noise variable.as we did for OUTPUT3.

OUTPUT7:

After Building Lasso Regression Model with noise variable positively correlated with the World Equities factor, as done in OUTPUT3 **and** λ=.1, we arrive at the following output:

```
Dependent Variable is Real Estate
Time period is between Febuary 2001 to September 2018 inclusive
lambda = 0.1
                    Intercept  World Equities  US Treasuries  \
LASSO Reg with noise  0.006142        0.806932       0.023147

                    Bond Risk Premium  Inflation Protection  \
LASSO Reg with noise               0.0                  0.0

                    Currency Protection  Noise
LASSO Reg with noise                 0.0    0.0
```

Depending on the noise, and the value of lambda, LASSO may shrink noise coefficient, or set it equal to zero. Let's try one last time with a larger lambda say lambda=0.2

OUTPUT8:

After Building Lasso Regression Model with noise variable positively correlated with the World Equities factor, as done in OUTPUT3 and λ=.2, we arrive at the following output:

```
Dependent Variable is Real Estate
Time period is between Febuary 2001 to September 2018 inclusive
lambda = 0.2
                     Intercept  World Equities  US Treasuries  \
LASSO Reg with noise   0.006777        0.709297            0.0

                     Bond Risk Premium  Inflation Protection  \
LASSO Reg with noise                0.0                   0.0

                     Currency Protection  Noise
LASSO Reg with noise                  0.0    0.0
```

Notice that for large lambda values, many coefficients have been set to zero, including the noise term.

CROSS VALIDATION

In the previous section we ignored the largest issue in LASSO regression, the choice of λ, In practice, and most people use cross-validation.

First, we will break the training set into k folds, and define a list of λ values. For each fold, and for each λ, we will train the model $k-1$ other folds, and calculate the error on the test fold.

At the end of this, we will have k out of sample errors for each value of lambda. Then we will pick the λ which satisfies producing the average error across our out of sample tests. Here we will use cross validation to pick the optimal lambda value for lasso.

OUTPUT9:

After Building Lasso Regression Model without noise and using cross-validation, we arrive at the following output with the best lambda value predicted by the model as 0.25.

```
Dependent Variable is Real Estate
Time period is between Febuary 2001 to September 2018 inclusive
best lambda = 0.25
          Intercept  World Equities  US Treasuries  Bond Risk Premium  \
CV Lasso   0.007021        0.662418            0.0                0.0

          Inflation Protection  Currency Protection
CV Lasso                   0.0                  0.0
```

OUTPUT10:

After Building Lasso Regression Model with noise variable positively correlated with the World Equities factor, as done in OUTPUT3 and using cross-validation, we arrive at the following output with the best lambda value predicted by the model as 0.25.

```
Dependent Variable is Real Estate
Time period is between Febuary 2001 to September 2018 inclusive
best lambda = 0.25
          Intercept  World Equities  US Treasuries  Bond Risk Premium  \
CV Lasso   0.007021        0.662418            0.0                0.0

          Inflation Protection  Currency Protection  Noise
CV Lasso                   0.0                  0.0    0.0
```

We can see that the cross validated LASSO model gives smaller factor loadings than OLS.Howver,Model Built using LASSO with and without Noise seem to not change much as Noise is getting completely Marginalized by LASSO

ELASTIC NET

Now that we've discussed cross validation and LASSO regression, we can mix and match penalized regressions to create regressions with specific properties. For instance, we know from literature that LASSO regression can be used for variable selection.

We also know that Ridge regression shrinks coefficients to provide a more robust solution. Combined, it's called an Elastic Net, and it can provide the benefits of both methods.

$$\hat{\beta}^{\text{LASSO}} = argmin_{\beta}\left\{\sum_{i=1}^{n}(y_i - X_i * \beta)^2 + \lambda_1 \sum_{j=1}^{m}|\beta_j|\right\}$$

$$\hat{\beta}^{\text{Ridge}} = argmin_{\beta}\left\{\sum_{i=1}^{n}(y_i - X_i * \beta)^2 + \lambda_2\|\beta\|_2^2\right\}$$

$$\hat{\beta}^{\text{Elastic Net}} = argmin_{\beta}\left\{\sum_{i=1}^{n}(y_i - X_i * \beta)^2 + \lambda_1 \sum_{j=1}^{m}|\beta_j| + \lambda_2\|\beta\|_2^2\right\}$$

In our notation, $\|\beta\|_2^2$ indicates the two norm of the vector β

Here, n is the number of data points, and m is the number of factors.

OUTPUT11:

After Building Elastic Net Model without noise , we arrive at the following output:

```
Dependent Variable is Real Estate
Time period is between January 1985 to September 2018 inclusive
best lambda1 = 0.04814799999999999
best lambda2 = 0.00024317171717171734
               Intercept  World Equities  US Treasuries  Bond Risk Premium  \
CV Elastic Net  -0.000348        0.311963       0.876042           0.835015

               Inflation Protection  Currency Protection
CV Elastic Net             0.047047                  0.0
```

OUTPUT12:

After Building Elastic Net Model with noise variable positively correlated with the World Equities factor, as done in OUTPUT3, we arrive at the following output:

```
Dependent Variable is Real Estate
Time period is between January 1985 to September 2018 inclusive
best lambda1 = 0.0655913
best lambda2 = 0.0003312691919191922
                          Intercept  World Equities  US Treasuries  \
CV Elastic Net with Noise   0.000434         0.32451       0.760312

                          Bond Risk Premium  Inflation Protection  \
CV Elastic Net with Noise           0.74916                   0.0

                          Currency Protection  Noise
CV Elastic Net with Noise                 0.0   -0.0
```

Here Elastic net did not do much better than LASSO because we don't have many highly correlated variables. If we had many highly correlated factors, we would expect the Elastic Net to outperform LASSO.

BEST SUBSET REGRESSION:

It attempts to find the linear model subject to the constraint that only "x" factor loadings can be nonzero. In this case, "x" is an integer the user defines.

Here, Let **z** be a vector of binary variables, let M be a very large number For simplicity, let total_vars be the number of variables considered and max_vars be the number max number of variables allowed in the subset.

$$\hat{\beta}^{\text{Best Subset}} = argmin_{\beta}\left\{\sum_{i=1}^{n}(y_i - X_i * \beta)^2\right\}.$$

$$\sum_{i=1}^{\text{max\_vars}} z_i \leq \text{max\_vars}, \quad Mz + \beta \geq 0 \text{ and } \beta \leq Mz, \mathbf{z} \text{ binary}$$

OUTPUT13:

After Building BEST SUBSET REGRESSION Model without noise, we arrive at the following output:

```
options['maxVars'] = 2
options['nameOfReg'] = 'Best Subset with maxVars = 2'
best_subset_regression(all_data, 'Real Estate',factorName, options)
```

```
Dependent Variable is Real Estate
Time period is between January 1985 to September 2018 inclusive
                             Intercept  World Equities  US Treasuries  \
Best Subset with maxVars = 2  -0.001956    2.103416e-09       1.453734

                             Bond Risk Premium  Inflation Protection  \
Best Subset with maxVars = 2           1.395798         -3.542204e-11

                             Currency Protection
Best Subset with maxVars = 2        -2.118101e-10
```

OUTPUT14:

After Building BEST SUBSET REGRESSION Model with noise variable positively correlated with the World Equities factor, as done in OUTPUT3, we arrive at the following output:

```
options['maxVars'] = 3
options['nameOfReg'] = 'Best Subset with maxVars = 3'
best_subset_regression(all_data, 'Real Estate', factorNameWithNoise, options)
```

```
Dependent Variable is Real Estate
Time period is between January 1985 to September 2018 inclusive
                             Intercept  World Equities  US Treasuries  \
Best Subset with maxVars = 3  -0.002295        0.274694       1.186268

                             Bond Risk Premium  Inflation Protection  \
Best Subset with maxVars = 3           1.09139         -1.454410e-10

                             Currency Protection         Noise
Best Subset with maxVars = 3       -1.101196e-11 -1.060274e-09
```