QRAFT

New Model of AI Driven Asset Management
Higher Alpha with Lower Cost through AI Technologies

QRAFT AI Quant Series

Factor Factory

Senior Manager, Sungmin Kim
Senior Manager, Eunchong Kim
Manager, Hanwook Jeong

June
2020

Contents

Qraft Technologies
+82 2 487 8555
qraft@qraftec.com

Summary

Financial factors have previously been discovered by the deductive method through research carried out by human researchers, and Factor Factory is a methodology that enables to automatically find the factors by the inductive method. Factor Factory uses Kirin, Qraft's proprietary financial data API, to access processed S&P Compustat data and to find suitable factor formula using advanced search algorithm. The found factor formula is evaluated its robustness and alpha by using various verification methods, including the rank IC. Until now, human researchers and the likes have been looking for the factors of the financial market. This method had the advantage of being easy to explain the calculated factor, but the process of finding the factor was difficult, and the study period took quite a long time. The purpose of Factor Factory is to make the process of factor discovery a task that computers can solve without human intervention by thoroughly quantifying the process, and to combine it with deep learning technology so the computers can find the factor efficiently.

Development Background

Most of the current research on factor investments are done by deducting formula using the economic implications of financial statement data. As meaning is given in a deductive way, most factor equations are not too complex and contain intuitive meanings. For example, the calculation processes of MKT, SMB and HML factors of the most commonly used Fama French 3-factor model in the stock market is not complex and only includes core meanings. The relatively recently published QMJ factor of Asness et al (2019) [1] has many financial elements used in the formula, but its complexity is not as complex. This deductive methodology, which produces formulas based on economic implications, has the advantage of being easy to interpret the meaning of the created factor formulas. However, the deductive approach is based on economic implications, so it takes a lot of time to discover a factor. At the same time, it is difficult to derive complex operations using various operators because formulas are derived by interpreting meanings.

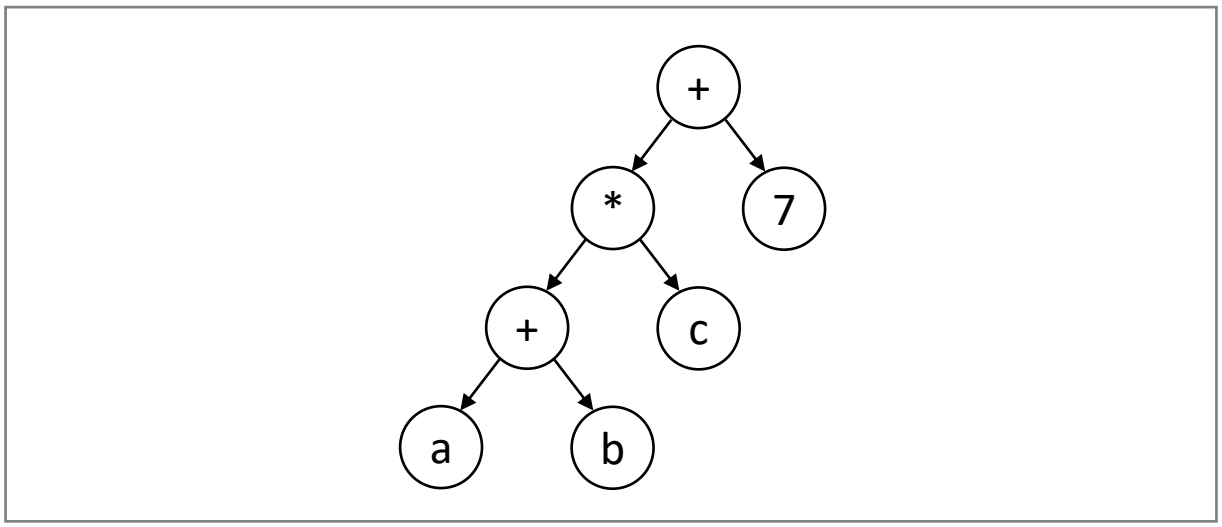
This is the main reason for developing Factor Factory. There are hundreds of financial data items available on S&P Compustat, and there are also many operators used to normalize and combine those data. Because the factor formula is determined by the combination of these two, it can be said that the number of factor formulas is virtually indefinite. Among them, the formula that can create alpha, which is difficult to find in conventional ways, will exist, even though in a small percentage, among meaningless expressions galore. The study attempts to explore the factor formulas that can create alpha in an automated way by systemizing the different methodologies to test the robustness and significance of alpha in factor investing and ways to create factor equations. Through this, the purpose of the research is to reduce the time it takes to discover the factor under the conventional method, while at the same time finding the factor with robust alpha under the deep complexity that has been difficult to find.

[1] Clifford S. Asness, Andrea Frazzini & Lasse Heje Pedersen. (2019). "Quality minus junk". Review of Accounting Studies, 24, 34-112

Basic components

Factor Factory developed by Qraft Technologies creates a factor formula with the available financial data and "Expression Tree" using multiple operators, and tests it by utilizing various indicators to explore the factor with robust alpha. To this end, there are four key elements of the universe configuration, factor tree creation, discovery algorithm and verification process, each of which is described below. [Figure 1] is a schematic diagram for the understanding of the "Expression Tree". An operator or number or variable is included in the tree branch. As for the example below, it represents "(a+b)*c+7" equation.

Figure 1. Example of Expression Tree



Source: Qraft Technologies

1. Universe

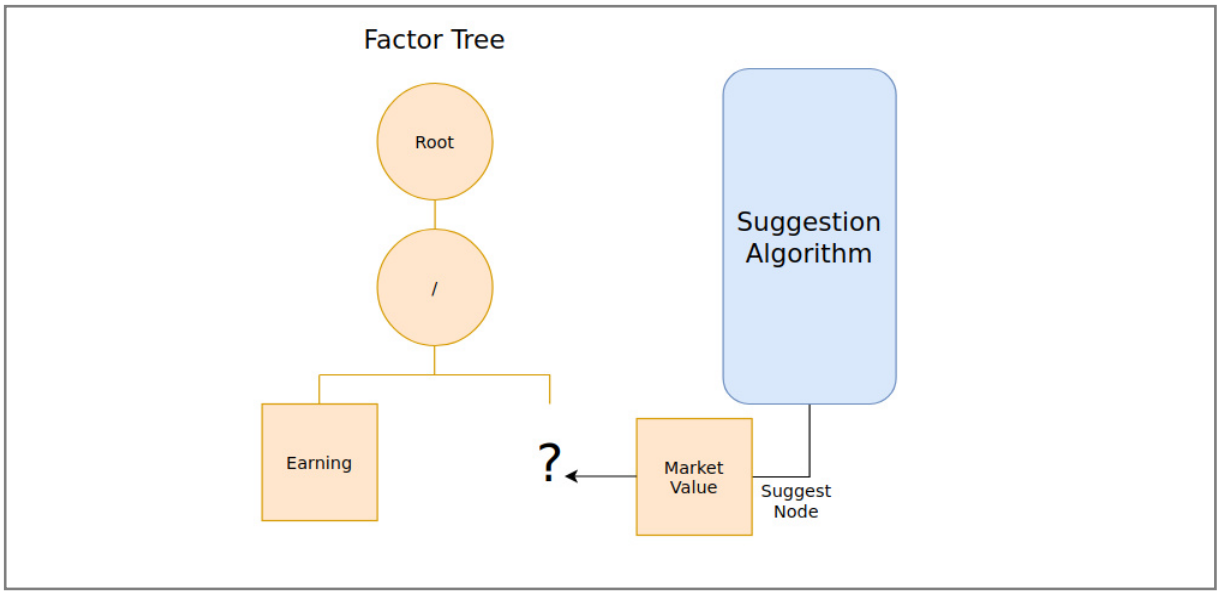
The first step is to declare an investment universe that will evaluate a factor. Even for the same factor, the effect can vary greatly depending on the universe setting and for this reason this is a simple but important step. Qraft's proprietary Kirin API enables effective management of investment universes and consists of the four settings below. It also reflects on a monthly basis all changes resulting from listing and delisting, in order to eliminate survival biases.

- ✓ Whether to only include class A stocks
- ✓ Which type of security to include (common stock, preferred stock, ADR, etc.)?
- ✓ Which stock exchange to include (NYSE, NASDAQ, AMEX, NYSE_ARCA, OTC, etc.)?
- ✓ Whether to set up minimum market cap standards

2. Factor Tree

A factor is expressed by an expression tree named “Factor Tree” as shown in [Figure 2]. Based on the status of “Factor Tree”, the specific discovery algorithm that will be explained later repeats the process of recommending a suitable node for the next position, and through this “Factor Tree” is completed. The nodes that can configure “Factor Tree” are divided into three types: operators, data, and normalizers. The operator nodes and normalizer nodes are alternately stacked up and the data node places on the stacked nodes in the end, constructing “Factor Tree”.

Figure 2. Factor Tree Structure



Source: Qraft Technologies

1) Operator Node

The operator node is a node used as an operator. A set of nodes consisting of various types of operators including the basic four fundamental arithmetic operations, shift operators and signed square root operators.

2) Normalizer Node

In the process of factor calculations, the scales of data used by operators are often significantly different, which can lead to a bias that relies heavily on specific data. Thus, to adjust this, the normalization methods such as Z-score and min-max are applied and, in some cases, the normalization process may be skipped.

3) Data Node

A set of nodes consisting of rate of return and financial data. Of the financial data items of Compustat, 111 data with a NaN value of 25 percent or less and price-related data such as price return, total return, and market capitalization constitute the data nodes. The data is collectively preprocessed using Qraft's Kirin API. In the process, only the shares included in the targeted universe are masked and look-ahead biases due to events such as restatements. All data exists in two types: raw data and data normalized by market capitalization.

3. Discovery Algorithm

The discovery algorithm completes "Factor Tree" by recommending a suitable node. The discovery algorithm completes the factor equation, presenting the node to be used in the factor formula sequentially. Factor Factory currently has two different discovery algorithms: the random search method and reinforcement learning method.

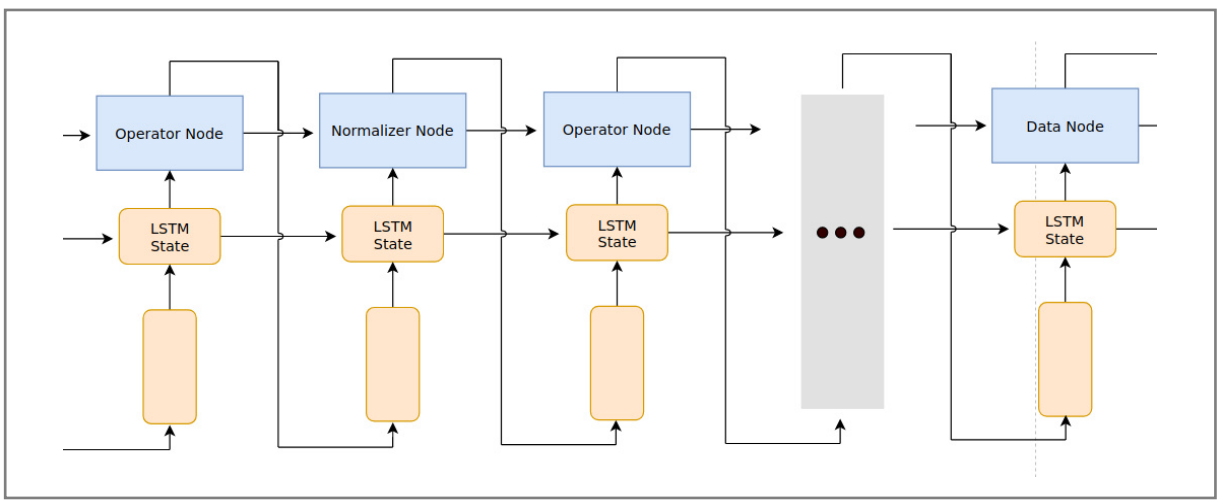
1) Random Search

The first method, random search, is to complete "Factor Tree" by recommending random nodes as one can infer from the name itself. Although the random search method is very simple, it often produces results that are compatible to other complex methods in many areas. In addition, unlike the reinforcement learning method described below, there is no overhead[2] of the search itself, so many "Factor Trees" can be completed in a short period of time. In the random search, you randomly select the nodes that can be applied to the factor tree.

2) Reinforcement Learning

The random search method has the above-mentioned advantages, but it does not explore a space efficiently. The number of branches of "Factor Tree", which is made from a combination of 111 financial data and operators, is so large that even if the depth of the tree is limited, it is impossible to explore the whole tree. In order to effectively explore a large space, a method other than the random search is needed, and various methods such as Bayesian Optimization can be used. The methodologies using reinforcement learning of deep learning, which have recently become prominent, were applied, and among them, the method was modified and improved to suit factor search based on Prajit et al (2017)[3] .

Figure 3. RNN Controller



Source : Qraft Technologies

The init node which shows the start status and data, operator, and normalizer nodes are embedded with node vectors, and RNN Controller in [Figure 3] receives embedded node vectors, presenting the next node to be filled in the factor tree. These presented node vectors are used as inputs in the next step and are used to present the next node. After all nodes are filled up to the data node at the end, the corresponding "Factor Tree" is evaluated to calculate the final two-dimensional factor matrix of [date, stock (securities)]. The verification process to be introduced later will quantify how well the formula for calculating the factor matrix is constructed and use it as a reward for reinforcement learning.

[2] Means indirect processing time, memory, etc. of a process
[3] Prajit Ramachandran, Barret Zopth, Quoc V.Le. (2017). "Searching for Activation Functions". Google Brain

Verification Process

The verification process is the process of evaluating the two-dimensional factor matrix of [date, stock (securities)] calculated by the factor formula. This is the step where evaluation on the robustness of alpha of the portfolio constructed by the corresponding factor matrix is carried out. The verification process consists of the process of forming a portfolio using the factor matrix and verifying the effectiveness of the factor.

1. Portfolio Construction

This step is the process of constructing a portfolio using a given factor. The composition of a portfolio can vary depending on whether long-short strategy is considered, how many shares are to be included in the portfolio, whether to set a market capitalization limitaton, and how the rebalancing period is determined. The calculation is done in parallel, taking into account the number of portfolio cases computed for the various conditions applied to the actual investing. Subsequently, the portfolio, of which the verification indicator, which is a combination of the values of the methodologies used for various factor verifications, maximizes is selected.

2. Verification Indicator

To verify the efficiency of the factor produced by Factor Factory, several verification methods are applied. The final verification indicator value for use in reinforcement learning consists of a combination of the following verification methods.

1) IC, Rank IC

To examine the validity of the factor, the information coefficient (IC) proposed by Grinold (1989)[4] is used. IC is calculated by the equation below.

$$IC_t = Correlation(Return_t, Factor\ Value_{t-1})$$

In other words, IC values refer to the correlation between the factor value at t-1 and the return on investment at t. In the IC value calculation process, when Pearson correlation that takes into account the rank of the values is used then it is defined as “Rank IC”. The higher the absolute value of the correlation between the factor value at t-1 and the rate of return at t, the greater the rate of return prediction effect of that factor. This is because if the IC value is high, a relatively high rate of return can be expected in the future as the investment is made on the stocks of the high/low factor value quantile. Therefore, the validity of the investment is verified through IC and Rank IC and used as a verification indicator of Factor Factory.

[4] Grinold, Richard C. (1989). “The Fundamental Law of Active Management.” Journal of Portfolio Management, vol. 15, no. 3(Spring) : 30-37

2) Mutual Information

Second, the quantity of mutual information, a characteristic of a probability variable that can replace a correlation coefficient, is used as a verification indicator. The amount of mutual information is measured from interdependence between two variables. The 'amount of information' obtained from one probability variable through observation of the probability variables is quantified. The equation is as follows.

$$Mutual\ Information(X,Y) = \sum_{i=1}^{N}\sum_{j=1}^{M} p(x_i, y_i)\ log\frac{p(x_i, y_i)}{p(x_i)\ p(y_i)}$$

If the two probability variables are completely independent, the value will be zero, and if the two are closely related, the value will be high. The amount of mutual information is equal to the difference between entropy[5] (H(X)) and conditional entropy (H(X|Y)).

$$Mutual\ Information(X,Y) = H(X) - H(X|Y)$$

$$H(X) = -\sum_{i=i}^{N} p(x_i)\ \log\{p(x_i)\}$$

Due to the characteristic that the greater the relevant information, the greater the value, and the smaller the relevance, the smaller the value, it is possible to compare the amount of mutual information each factor has. As with the IC above, the quantity of mutual information between the factor values and the rate of return determines the effect of predicting the rate of return of the factor and is utilized as the verification indicator.

3) CAPM / FF3F Alpha

To have a look at the excess rate of return of the factor, we analyze market beta (β), which is the systematic risk of CAPM, and excess rate of return which is derived from controlled MKT (Market), SMB (Small Minus Big), and HML (High Minus Low), which are three factors of Fama-French (FF3F, 1993[6]) .

$$CAPM : R_p - R_f = \alpha + \beta_1(Mkt) + \varepsilon$$

$$FF3F : R_p - R_f = \alpha + \beta_1(Mkt) + \beta_2(SMB) + \beta_3(HML) + \varepsilon$$

The α in the equation represents the excess return rate of the portfolio, MKT represents the market premium, SMB represents the enterprise-scale premium, and HML represents the book value/market value premium. The ε in the equation represents error term, and Rf represents risk-free return and uses the call rate. In addition, to take into account additional statistical significance, we check if the p-value is less than 0.05. Through this, CAPM, a simple linear regression model, and FF3F, a multi regression model, are used as verification indicators to confirm whether they have generated excess revenue.

[5] Entropy is the expected amount of information about the overall state and is the maximum when each of the N states has a probability of 1/N

[6] Fama, E. F., French, K. R. (1993). "Common risk factors in the returns on stocks and bonds." Journal of Financial Economics. 33:3-56

4) GRS Test

To test the robustness of the factor, we use the GRS F-Test proposed by Gibbons, Ross, and Shanekn (1989)[7]. This test examines whether the mean-variance curve of the portfolio and the maximization of the Sharpe ratio are statistically significant. The GRS F-Test validates the null hypothesis that the constant term of the entire regression equation is zero, which tests the validity of the comparative model by testing whether the Jensen's alpha in each portfolio is statistically zero, instead of testing the statistical significance of the individual beta.

$$F - value = \left(\frac{T}{N}\right)\left(\frac{T-N-L}{T-L-1}\right)\left(\frac{\widehat{\alpha'}\,\widehat{\sum^{-1}}\,\hat{\alpha}}{1+\overline{\mu}'\hat{\omega}\overline{\mu}}\right) \sim F(N, T-N-L)$$

$\hat{\alpha}$: Estimated constants term ($N \times 1$ matrix)

$\overline{\mu}$: Factor matrix ($T \times L$ matrix)'s sample mean($L \times 1$ matrix)

$\hat{\omega}$: Factor matrix($T \times L$ matrix) covariance matrix unbiased estimate($L \times L$ matrix)

$\widehat{\sum}$: Residual covariance matrix's unbiased estimate ($N \times N$ matirx)

The above equation is the formula of GRS F-Test. If the verification results of the GRS F-Test fail to reject the null hypothesis, it indicates that the comparative model explains the verification object portfolio return. On the other hand, if GRS F-Test rejects the null hypothesis, it indicates that the comparative model cannot fully explain the return on the portfolio to be verified. The verification of the factor portfolio is carried out using the FF3F model, which is a typical pricing model. If the GRS F-Test validation results fail to reject the null hypothesis, the return on the factor-based portfolio is explained by the model. On the other hand, if the null hypothesis is rejected, it means that there is excess return (α) of the factor that has been discovered by Factor Factory, which cannot be explained by the comparative model, FF3F. Whether to reject this null hypothesis determines the validity of the excess revenue of the factor and is used as a verification indicator of Factor Factory.

5) Mean-Variance Spanning Regression

The Mean-Variance Spanning Test of Huberman and Kandel (1987)[8] is used as a way to verify the efficiency of the portfolio in terms of mean-Variance. We evaluate whether the return on an existing portfolio can be "spanned" by adding the return of the portfolio derived by the factor values to the benchmark portfolio through regression analysis. We assume that the return of the portfolio with additional factors as a dependent variable and the return of the benchmark portfolio as an independent variable, and the regression equation is as follows:

$$r_t = \alpha + \beta R_t + \varepsilon t$$
$$H_o : \alpha = 0 \ and \ \beta = 1$$

r_t : Rate of returns of the portfolio with the additional factors at t

R_t : Rate of return of the benchmark portfolio at t

When rejecting the null hypothesis is impossible, factor return can be spanned by the existing benchmark return, meaning that adding a factor to the existing benchmark portfolio and constructing a new portfolio would not improve the performance in terms of rate of return. Conversely, if the null hypothesis can be rejected, it means that performance in terms of rate of return can be improved. In addition, the value of α is utilized at this time, and the F test determines whether to reject the null hypothesis.

[7]Michel R. Gibbons, Stephen A.Ross, Jay Shanken. (1989). "A Test of The Efficiency of a Given Portfolio." Econometrica, Vol. 57, No. 5, 1121-1152

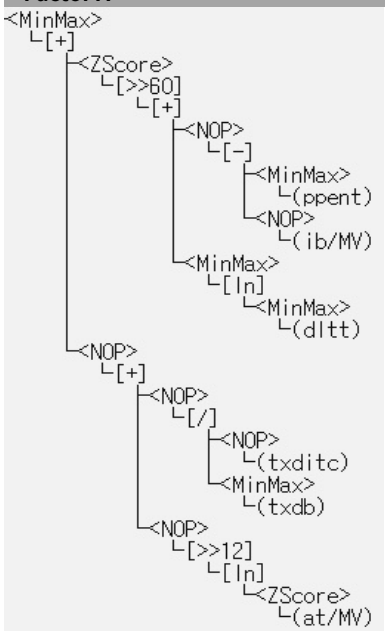
[8]Gur Huberman, Shmuel Kandel.(1987) Mean-Variance Spanning. Journal of Finance, vol. 42, issue 4, 873-88

Experiment Result

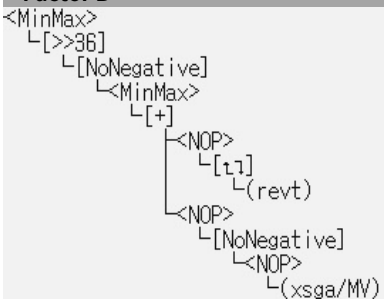
[Figure 4] shows some of the factors found by Factor Factory. In the course of the experiment, data over the last five years were not used in learning. In “Factor Tree”, parentheses are data nodes expressed by the financial data names on S&P Compustat, and brackets are operator nodes that perform operations based on the data. Square brackets mean the normalize nodes that proceed with the normalization of the value. Details of each node that constitutes a factor equation are included in the appendix.

Figure 4. Factor Tree

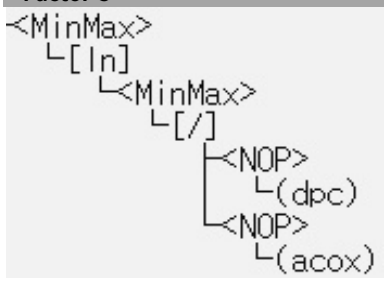
Factor A



Factor B



Factor C



Source: Qraft Technologies

The portfolio construction using each factor was done with a one-month rebalancing period and a market cap-weighted method. [Table 4] shows the performance of the above factors, and since no data of the last five years has been used in the experiment, it is possible to compare the actual performance of the corresponding factors as out-of-samples. The rate of return and Sharpe ratio of each factor are superior to the benchmark, and the same pattern is shown in in-samples, just as the case for out-of-samples. This suggests that the factor is not affected by overfitting in the course of learning and is generating robust excess return.

Table 4. Performance Summary of Factor Tree

	BM	Factor A	Factor B	Factor C
# of Stocks	-	50	100	200
Panel A: In Sample(Train period) 1992-01-31~2015-04-30				
Mean return	0.0066	0.0114	0.0102	0.0095
Std.D	0.0414	0.0516	0.0510	0.0469
Sharpe	0.1603	0.2210	0.1998	0.2027
Panel B: Out of Sample(Test period) 2015-05-31~2020-04-30				
Mean return	0.0065	0.0102	0.0081	0.0089
Std.D	0.0421	0.0528	0.0440	0.0495
Sharpe	0.1539	0.1932	0.1836	0.1788

Source: Qraft Technologies, Compustat

[Figure 5], [Figure 6] and [Figure 7] below are graphs showing the performance of Factor A, Factor B and Factor C respectively. The chart also shows the rate of return of S&P 500, the benchmark index, and excess rate of return compared to the market.

Figure 5. Factor A Portfolio Performance Chart

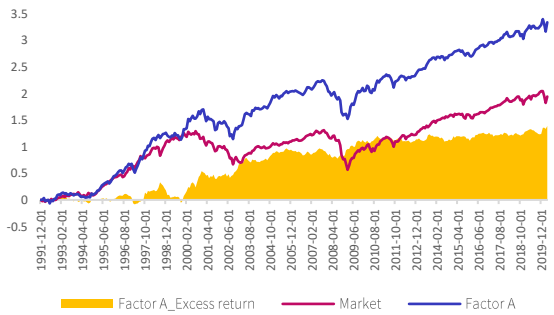


Figure 6. Factor B Portfolio Performance Chart

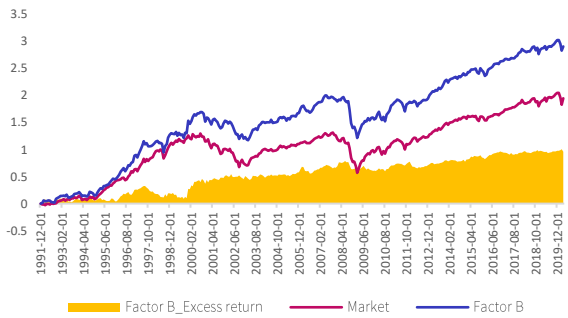
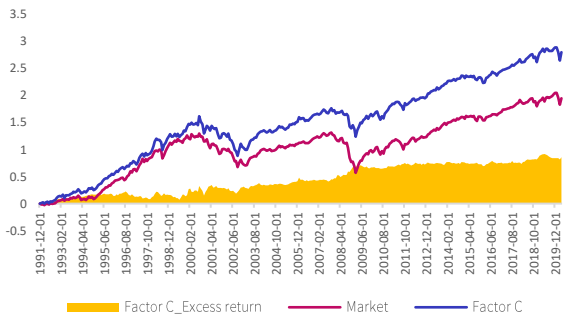


Figure 7. Factor C Portfolio Performance Chart



Source: Qraft Technologies, Compustat

Conclusion

Using the inductive factor search method of Factor Factory, it is possible to find a factor that is difficult to identify with the conventional deductive method. In addition, the speed of factor discovery is noteworthy. Even with a personal computer, you can explore more than 10 significant factors a day (based on CAPM monthly alpha 0.001 or higher, on a significant level of 5%). As such, Factor Factory shows the possibility of automated factor-finding by creating a robust alpha that is actually utilizable.

On the other hand, there are limitations of the current Factor Factory. One of the limitations is low explanatory power. What the derived Factor Tree means economically can only be understood when the user interprets on his/her own. However, it is not easy to interpret a random factor formula in an economic way, and it is easy to interpret it in an arbitrary way. If the experimentation process and the test process are thoroughly separated and the factor is verified to be statistically significant meticulously, there is no problem in determining the significance of alpha, even without the economic implications. However, if you are a person who prefers only what you can understand, you will have qualms about using the derived factor. The second is that we are not considering the relationship between the calculated factors. Considering that reinforcement learning searches the Factor Tree in an inductive way, it is highly likely that it will be fitted toward the factor formula that has already been found, or that sub-trees of the Factor Tree will continue to be recycled. This increases the correlation between the factors, thus sharing the same unsystematic risks. Due to this, the future Factor Factory will be aimed at exploring a group of factors, not at a single factor level, and will be designed to maximize the alpha of a factor group rather than a single factor.

Factor Factory is a project that is currently at the early-stages, and there are still many areas that can be upgraded. Therefore, if we overcome the above limitations and develop Factor Factory, we will be able to bring numerous independent alphas hiding in the financial world to the surface.

Appendix

Information on the nodes used in the experiment

1. Normalizer Node

1.1 No-Operation [NOP] : Does not Normalize and Skip
1.2 Z-score normalize [ZScore] : Cross sectional Z-score Normalize
1.3 Min Max normalize [MinMax] : Cross sectional Min Max normalize

2. Operator Node

2.1 Add [+] : Retrieve two data matrixes and add.
2.2 Subtract [-] : Retrieve two data matrixes and subtract.
2.3 Safe-Divide [/] : Retrieve two data matrixes and divide. In the process of division, add an epsilon to avoid zero-division error.
2.4 Shift-N [>>N] Retrieve a data matrix and shift by N-months.
2.5 Signed-SafeLog [ln] Retrieve a data matrix and add 1 and take the absolute value and log it. If the retrieved data is a negative value, multiply minus 1.
2.6 Invalidate Negative [NoNegative] Retrieve a data matrix and remove values less than 0.
2.7 Upside-down [↥↧] Retrieve a data matrix and rotate the locations based on the mean.

3. Data Node

All data values are post processed based on S&P Compustat data. If /MV is attached to the back of the data node, it means that the data are divided by the corresponding market capitalization.

3.1 acox : Current Assets - Other - Sundry
3.2 at : Assets - Total
3.3 dltt : Long-Term Debt - Total
3.4 dpc : Depreciation and Amortization (Cash Flow)
3.5 ib : Income Before Extraordinary Items
3.6 ppent : Property, Plant and Equipment - Total (Net)
3.7 revt : Revenue - Total
3.8 txdc : Deferred Taxes (Cash Flow)
3.9 txditc : Deferred Taxes and Investment Tax Credit
3.10 xsga : Selling, General and Administrative Expense