CSE514A Datamining – Fall 2018 Final Report

Empirical Asset Pricing

Introduction

Our project topic is empirical asset pricing. Given characteristics variables of a company, we are interested in building models for stock return prediction or classification. With the repertoire of methods like generalized linear models, boosted regression trees, random forest and neural networks, we implemented Principal Component Regression, Gradient Boosted Regression Trees and Neural Networks.

The main challenge here is that we have a broad set of characteristics for each company: they could derive from the security files like max daily return and momentum, come from fundamentals of the company, like dividend to price ratio and debt to equity ratio. Since many of the variables are strongly correlated, it will be a good idea to use methods which have less 'restrictions' to the variables.

Therefore, Neural Networks method becomes our first choice, which is able to reveal the unseen correlation between two correlated variables. Principal component regression also works reasonably well, since it can help reduce irrelevant features. GBRT model builds an additive model in a stage-wise fashion; it allows for the optimization of arbitrary differentiable loss functions. In each stage a regression tree is fit on the negative gradient of the given loss function. With GBRT, we can handle heterogeneous features. We also benefit from ensemble strategy of boosting so that we can have a robust final model against outliers in output space.

Existing Methods

Among those existing methods, the least complex one is the simple linear predictive regression model, which can be estimated by the ordinary least squares (OLS). Although we expect this method to perform poorly in our case with high dimensions, we use this method as a reference to emphasize distinctive features of our more sophisticated methods. So firstly, we could have a simple predictor, then we can compute the error residual and learn to predict that residual before combining a better predictor. GBRT builds an additive model in a stage-wise fashion. We benefit from the ensemble strategy of boosting so that we can have a robust final model against outliers in output space. PCR is built upon PCA which is used for dimension reduction given a large number of features.

Methodology

Our main objective is:
Given dataset of m data samples, each data sample has n_x dimensions(predictors) and dependent variables. In other words, we haves a single m samples of (x1,y),(x2,y),(xi,y)...(x_{nx},y) where xi's are the predictors such as change of common shares outstanding and y's are the actually stock closing price. The objective is to make predictions of y given a new independent variable X = (x1,x2...x_{nx}) and to minimize the error or Cost Function of each our model. Specifics are to be discussed respectively in following chapter.

Models

We decide to use 3 methods among the previously proposed methods given our problem. The methods are Principal Component Regression, Gradient boosted regression trees and Neural networks and Random Forest Regressor.

PCR (Principal Component Regression)

As we know, if we have relatively too many features *p* of a data sample compared to the number of data samples, we may encounter the "curse of dimensionality ". In other words, we would have overfitting problem because we have so many variables in order to predict stocks. And the model is likely not going to work especially and "troublesome for the problem of return prediction where the signal-to-noise ratio is notoriously low"[1] where *SNR=x(mean)/***s.**

So, there are several ways of handling high dimensionality. As analyzed by Gu, one way is "Penalized Linear" which introduces a parameter for penalty. For example, we can use LASSO. However in our case, it is considered a better approach to use an ensemble strategy, such as predictor averaging. The idea of predictor averaging is the key point of dimension reduction. By forming linear combinations of predictors, we can reduce the noise in order to isolate the signal in predictors better. If two predictors are highly dependent, by doing son, we can also de-correlate these two. And the main dimension reduction techniques that we use here is Principal Components Regression (PCR).

We take linear regression model from previous proposal $r_{i,t+1} = E_t(r_{i,t+1}) + \varepsilon_{i,t+1}$ and and perform vectorization. So we have following as our model:

$$R = Z\theta + E;$$

where R is the NT x 1 vector of $r_{i,t+1}$, Z is the NT x P matrix of stacked predictors $z_{i,t}$, and E is a NT x 1 vector of residuals $\varepsilon_{i,t+1}$.

and we condense dimension from P to K:

$$R = (Z\Omega K)\theta K + E;$$

[1] Shihao Gu, Bryan Kelly and Dacheng Xiu, 2018, Empirical Asset Pricing via Machine Learning, pp11

PCR chooses combination weights ΩK recursively and our for jth linear combination solves:

$$w_j = \arg\max_{w} \mathrm{Var}(Zw), \quad \text{s.t.} \quad w'w = 1, \quad \mathrm{Cov}(Zw, Zw_l) = 0, \quad l = 1, 2, \ldots, j-1. \qquad [1]$$

There are two steps used in the PCR model. First step is the principal component analysis, in which we combine regressors into a small set of linear combinations that form the covariance structure among the predictors. Second step is using several leading components in standard predictive regression. By using this PCR model, we could clear the coefficients of those higher order components, so that regularize the prediction problem. Since the 25 variables related to stock prices are highly correlated, so this model helps us pick variables that have eigenvalues bigger than one, giving us 6 leading components.

The following is the progress result:

```
Call:
lm(formula = price ~ ., data = as.data.frame(modeldata))

Residuals:
    Min      1Q  Median      3Q     Max
-49.730  -9.274  -0.215   8.781 114.552

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 24.4983388  0.7521199  32.572   <2e-16 ***
PC1         -0.0036024  0.0001292 -27.874   <2e-16 ***
PC2         -0.0780497  0.0014719 -53.027   <2e-16 ***
PC3         -0.0234776  0.0006537 -35.915   <2e-16 ***
PC4         -0.0123602  0.0008413 -14.691   <2e-16 ***
PC5          0.0156160  0.0011511  13.567   <2e-16 ***
PC6          0.0233474  0.0005004  46.658   <2e-16 ***
PC7         -0.0001120  0.0007721  -0.145    0.885
---


Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
Residual standard error: 15.03 on 1507 degrees of freedom
Multiple R-squared:  0.8314, Adjusted R-squared:  0.8307
F-statistic:  1062 on 7 and 1507 DF,  p-value: < 2.2e-16

[1] 86.98341
```

[1] Formula- Shihao Gu, Bryan Kelly and Dacheng Xiu, 2018, Empirical Asset Pricing via Machine Learning, pp13

After running the model on our dataset, the calculated multiple R-squared is 0.8317 and the adjusted R-squared is 0.8307. The data shows that 83% of the variation in the variables is explained by my model.

GBRT(gradient boosted regression trees)

Another idea of building our model is that not only "predictor's non-linear impact on expected returns", we should also "account for interactions among predictors"*[1]*. For example, consider we have a regression which is:

$$y = w^T X+b$$

There might be some interactions like:

$$y= w^T X +\alpha XX^T + b$$

How to address or simulate this fact? Aside from utilizing neural networks with multiple layers, an alternative is to use regression trees. Regression trees are used when our goal is making numeric predictions instead of doing classification, compared to classification trees. And they are easy to implement with recursion.

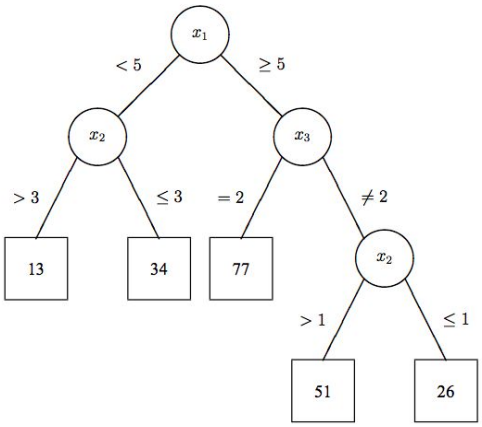


figure A regression tree example [2]

[1] Shihao Gu, Bryan Kelly and Dacheng Xiu, 2018, Empirical Asset Pricing via Machine Learning, pp15
[2]Breiman, Leo, Jerome Friedman, Charles J Stone, and Richard A Olshen, 1984, Classification and regression trees (CRC press), p258 ,Chap11

It recursively makes partitions of X into smaller regions. We start with all the data at root node and apply regression formulas for every partitions of every feature. So, if we have (xi , yi),(x2, y2), . . .(xc, yc) in leaf node k, the model for k is $\hat{y} = 1/c \sum i=1$ to c yi, which is just the mean of dependent variable in this partition. The formal model is as follows:

$$g(z_{i,t}; \theta, K, L) = \sum_{k=1}^{K} \theta_k \mathbf{1}_{\{z_{i,t} \in C_k(L)\}},$$

where Ck(L) is one of the K partitions of the data. Each partition is a product of up to L indicator functions of the predictors. The constant associated with partition partition k (denoted k) is de ned to be the sample average of outcomes within the partition.

[1]

At each split, we will compute the Mean Squared Error(MSE) per sample(Formula listed below) and we associate θ with the larger of the MSE. We select feature that has minimum MSE value.

$$H(\theta, C) = \frac{1}{|C|} \sum_{z_{i,t} \in C} (r_{i,t+1} - \theta)^2,$$

[2]

We will stop recursion when there are less than k nodes and if retained minimum MSE is smaller that our specified threshold. Variables and progress will be discussed in following report.

We decide on the number of iterations for the gradient boosted trees by comparing the

Two-layer Neural networks

SVM is more for classification. Based on Gu's statistics, "Allowing for nonlinearities substantially improves predictions".[3] So another qualified and powerful method we consider is neural networks which allows interactions between predictors and takes advantage of non-linear transforms from activation functions. It is also suggested by GU that shallow NN out performs deep NN. And we decide to have two layers which is a hidden layer and a output layer.

For our model, we just use traditional neural networks covered in the course lecture. For each data sample, the input layer consists the predictors x1,x2,...xp. In hidden layer, each neuron will be an

---

[1] Shihao Gu, Bryan Kelly and Dacheng Xiu, 2018, Empirical Asset Pricing via Machine Learning, pp15
[2] Shihao Gu, Bryan Kelly and Dacheng Xiu, 2018, Empirical Asset Pricing via Machine Learning, pp16
[3] Shihao Gu, Bryan Kelly and Dacheng Xiu, 2018, Empirical Asset Pricing via Machine Learning, pp5

activation function of a linear regression. For typical activation functions of our model, we will use rectified linear unit (ReLU), which is :

$$\text{ReLU}(x) = \begin{cases} 0 \text{ for } x<0 \\ x \text{ for } x>=0 \end{cases}$$

And for output layer, we will not have activation function because will will output real numbers of stock price. And we will use gradient descent. In other words, use chain rule to back propagate and calculate derivatives of dL/dw and set it as gradient to update parameters such as w = w-dL/dw to minimize Loss in SGD.

Formally, Our neural network model has the following general formula. Let K(l) denote the number of neurons in each layer l = 1...L. Define the output of neuron k in layer l as $xk^{(l)}$ Next, define the vector of outputs for this layer (augmented to include a constant, $x0^{(l)}$) as x(l) = (1, $x1^{(l)}$, …, $xk^{(l)}$)'. To initialize the network, similarly define the input layer using the raw predictors, $x^{(0)}$ = (1,z1, ... zN)'. The recursive output formula for the neural network at each neuron in layer l > 0 is then

$$x_k^{(l)} = \text{ReLU}\left(x^{(l-1)'}\theta_k^{(l-1)}\right)$$

with output:

$$g(z;\theta) = x^{(L-1)'}\theta^{(L-1)}.$$

We estimate the neural network weight parameters by minimizing the penalized l2 objective function of prediction errors.

[1]

Dataset, Pre-processing and Meaningful Variables

We gathered raw data from CRSP & COMPUTESTAT. Then through some research and testing, we eventually calculated 35 variables as a combination for our asset pricing models. For example, max daily return is calculated through queries of daily return and a map-reduce job.

Dataset

We accessed CRSP & COMPUTESTAT via school's subscription of Wharton Research Data Services. And comes with the interval of a quarter. Our raw data comes from a list of 20 Fortune 100 industrial companies (ticker indicated in braces):

[1] Shihao Gu, Bryan Kelly and Dacheng Xiu, 2018, Empirical Asset Pricing via Machine Learning, pp20

Ford(F); General Motor(GM); General Electricity(GE); 3M(MMM); Caterpillar(CAT); EMERSON Electric(EMR); Honeywell(HON); Danaher(DHR);United Technologies Corporation(UTX);United Technologies Corporation(LMT); General Dynamics Corporation(GD); Raytheon Company(RTN); Northrop Grumman Corporation(NOC); Deere & Company(DE); Illinois Tool Works Inc.(ITW);Waste Management, Inc.(WM); Eaton Corporation plc(ETN); Roper Technologies, Inc.(ROP); CRH plc(CRH);Ingersoll-Rand Plc(IR)

The rationale for choosing this group of companies is that they are relatively more predictable than smaller companies or technology companies. They are less susceptible to changes in investor confidence or sudden, temporary movement in the market.

However, as it has been stated in proposal, we need more predictors more complex models other than preliminary time series regressions. We discussed and acquired some meaningful characteristics which will be addressed in following session.

Pre-processing

We accessed variables of companies' characteristics from CPSP and Computestat IQ. However, these are raw data that needs pre-processing. The formulas are already indicated above and are specific to the variables we have access to. After calculating the variables we need from raw data, we did further process:

- Fill data which is missing significantly. For example, calculate market value with stock price x common share outstanding.
- Uniform data collected to quarterly data. For example, use yearly book value and insert into quarterly table and use monthly stock close price to calculate market value.

Raw variables recorded quarterly are listed as following(from CRSP & COMPUTESTAT):

---

fundamentals:
CSHOQ -- Common Shares OutstandingACTQ -- Current Assets - ANCQ -- Non-Current Assets - Total TotalCAPXQ -- Capital ExpendituresCH -- CashREVTQ -- Revenue - TotalMKVALTQ -- Market Value - Total - FiscalUDPFA -- Depreciation of Fixed AssetsEPSPXQ -- Earnings Per Share (Basic) Excluding Extraordinary ItemsBKVLPS -- Book Value Per ShareDLTTQ -- Long-Term Debt - TotalINVTQ -- Inventories - Total DPQ -- Depreciation and Amortization - Total XRDQ -- Research and Development Expense PPEGTQ -- Property, Plant and Equipment - Total (Gross) - Quarterly

Note: bkvlps is recorded yearly

security:
PRCCQ -- Price Close - Quarter PRCCM-- Price Close - Month

Meaningful characteristics

As stated in the project proposal, it is not sufficient to make predictions on securities based on past stock price. Instead, we gathered information on characteristics of each of companies we chose and did some calculation. Although it might be useful to consider variables related to the overall context, the company's own characteristics are much more significant in terms of changing the performance of its security.

- So for PCR(first method we will use), the variables that we eventually use are the following:

    1. **chcsho:** Change in shares outstanding-compute
       chcsho = revtq/cshoq x prccq
    2. **agr:** Asset change of non current+current
       agr = (actq+ancq).diff()
    3. **rd_mve**: R&D to market capitalization
       rd_mve = xrdq/mkvaltq
    4. **invest**: Capital expenditures and inventory
       invest = capxy+invtq
    5. **cashpr:** cash productivity(dropped due to insufficient data)
       ch/revt
    6. **ep:** Earning to price (E/P ratio)
       ep = epspxq/prccq
    7. **depr**: Depreciation / PP&E (Property, Plant And Equipment)
       depr = dpq/ppegtq
    8. **bm**: Book-to-market Book value/share/stock price
       bm = bkvlps/prccq
    9. **lgr:**Growth in long-term debt
       lgr = dlttq.diff()
    10. **chinv:** Change in inventory
        chinv=invtq.diff()
    11. **mom1q:** momentum of 1 quarter
        mom1q= prccq/prccq(1 quarter before) x 100
    12. **mom2q:** momentum of 2 quarter
        mom2q=prccq/prccq(2 quarters before) x 100
    13. **mom1y:** momentum of 1 fiscal year
        mom1y=prccq/prccq(4 quarter before) x 100

- So for other methods, we added more variables based on measuring the performance

    14. **sp:** Sale to price
        sp=REVTQ/CSHOQ x PRCCQ
    15. **mvel1:** size

mvel1=mvel1

16. **dvpspq:** dividend per share
    dvpspq=dvpspq
17. **cshtrq:** Common Shares Traded / Quarter
    cshtrq=cshtrq
18. **turn:** Share turnover
    turn=saleq
19. **dolvol:** Dollar trading volume
    dolvol=cshtrq x prccq
20. **mom36m:** momentum of 36 months
    mom1y=prccq/prccq(12 quarter before) x 100
21. **maxret:** max daily return
    maxret= VWRETX-mapReduce()
22. **spread:** Bid-ask spread
    spread=bid-ask
23. **actq:** Current Assets - Total
    actq=actq
24. **ancq:** Non-Current Assets - Total
    ancq=ancq
25. **cshoq:** Common Shares Outstanding
    cshoq=cshoq
26. **dlttq:** Long-Term Debt - Total
    dlttq=dlttq
27. **invtq:** Inventories - Total
    invtq=invtq
28. **ppegtq:** Property, Plant and Equipment / Total (Gross) / Quarterly
    ppegtq=ppegtq
29. **revtq:** quarterly revenue
    mom1y=prccq/prccq(4 quarter before) x 100
30. **xrdq:** Research and Development Expense
    mom1y=prccq/prccq(4 quarter before) x 100
31. **capxy:** Capital Expenditures
    capxy=capx**y**
32. **mkvaltq:** Market Value - Total
    mkvaltq=mkvaltq
33. **dpq:** Depreciation and Amortization - Total
    dpq=dpq
34. **epspxq:** Earnings Per Share (Basic) / Excluding Extraordinary Items
    epspxq=epspxq
35. **bkvlps:** Book value per share
    bkvlps=bkvlps

Metrics and Comparison Results

Main result
The following chart is the main testing result for our methods. It is a classification problem so we do not have results without ground truth. We performed random forest in addition to GBRT.

Metrics
Our main metrics for comparison are:
1.R - Squared
2.Mean Square Error (Used for the comparison between same method but different type)
3.Time complexity

In addition to the comparison of the methods, we ranked feature importance with:
1.F-test
2.Mutual Information
3.Feature ranking with GBRT and Random Forest

Methods/Matrics	r2	MSE	Complexity
PCR	0.830	1322.96	$O(p^2 n+p^3)$
3NN	0.67	1703.58	O(n H)
GBRT	0.562	767.71	O(K d n logn)
RF	0.464	938.04	O(K m n logn)

* for time complexity: p is number of features; n is number of observations; H is sum of products of adjacent layers in NN, the first layer being the number of features. Eg p x h1+ h1 x h2...; K is total number of trees; d is max depth of tree; choose m features out of p features in RF.

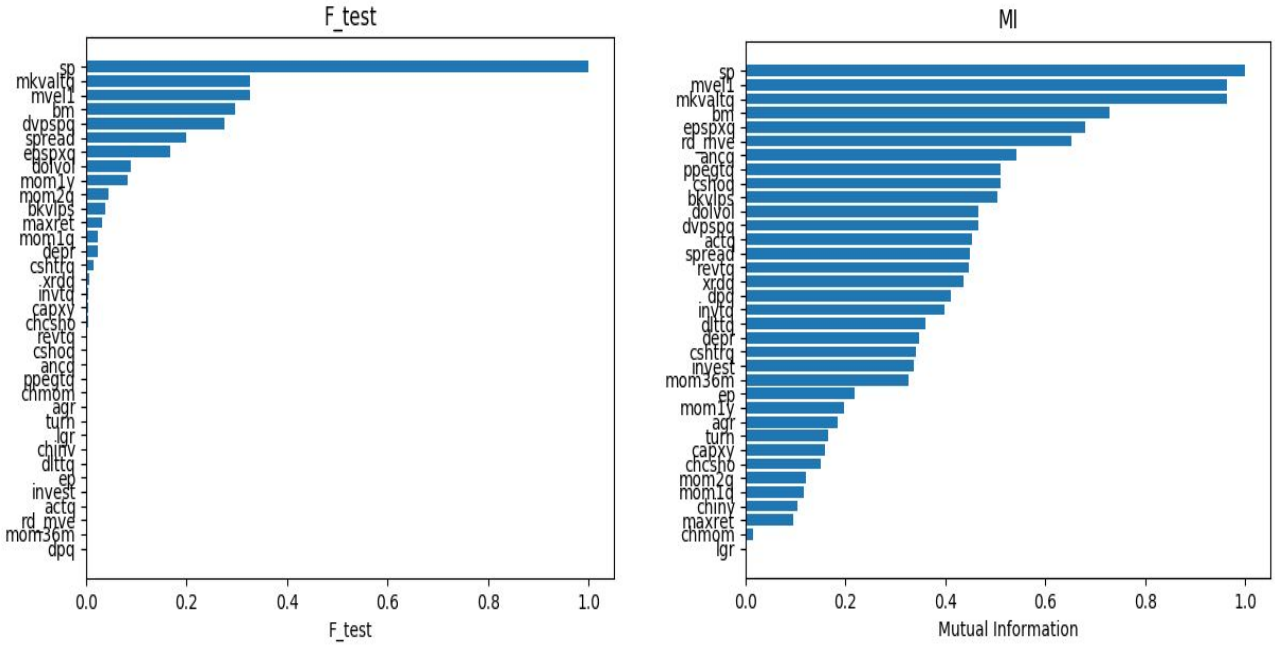
*data derived from assigning 90% of data as training set and 10% as test set.
*PCR and 3NN have different variables with GBRT and RF.
*there are around 4000 rows of data and 35 features concluded from last chapter

We can see that the generalized linear has good r^2 . In our case, it is possible for certain variable(s) to have strong linear relationship with the target althoug there can be non-linear impact of the predictor. But in this case, PCR out performs NN. Also, considering the fact some interactions among predictors(stated last chapter in GBRT), we can see that GBRT has good performance given many features.

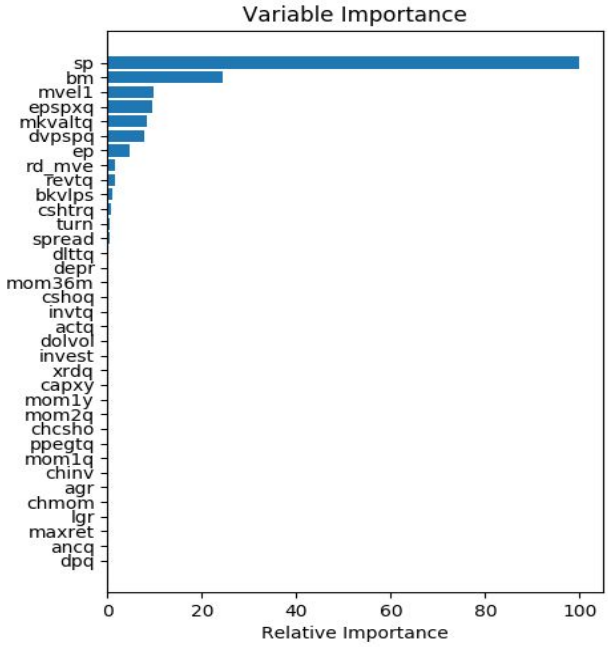
In terms of time complexity. All of the methods have polynomial complexity lower thant $O(n^2)$, given the case n>>p that the number of data samples is the nominant term. So the complexity is not dramatically different. However, if the number of features grows, say $n \approx p^2$, PCR will become more time-consuming as opposed to other methods. In addition, we used shallow neural networks. And if there is a deep NN, the time complexity will grow dramatically.

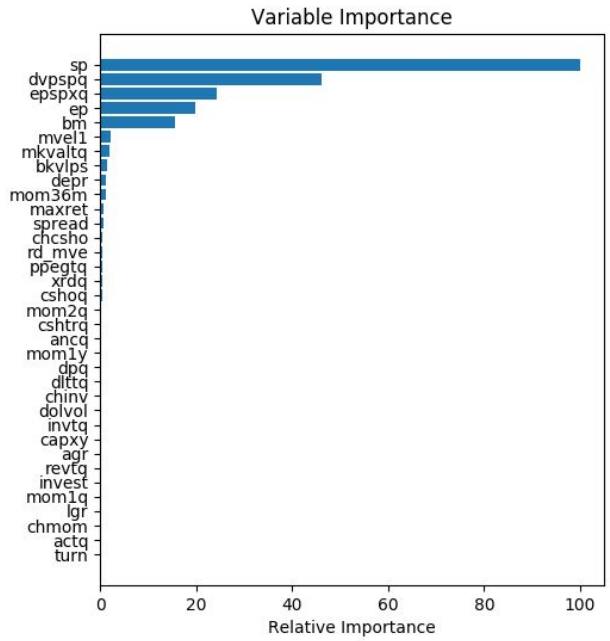
However, the idea of our project is not only evaluating the result. More importantly, it is crucial that we work back and forth to build a certain charateristic set that can have explanation ability in the given context. So we did some research and tried out the variables. Then we used multiple ways to measure the importance of the variables.

With F_test we are able to capture the linear dependency between variable and the target. With a step forward, Mutual Information(MI) can capture any kind of dependency between variables. The result is the following:



More over, GBRT and Random Forest have intrinsic ability of feature ranking. We generated the feature ranking from these methods which gives use another angle to view the charateristices. We can see that although in studies of larger number of companies and in more generalized industries momentum is considered a vary important characteristic, in our specific case of industry companies, there are some other features that are most important, like sale to price. Our conclusion here is when determining the the most powerful charateristics, it is indispensable to consider within the context of the industry. It is also feasible to label the company and build models suitable for the companies of the same label.





The graph above demonstrates the importance of different variables in determining asset price. Here, relative importance is thought of as the loss in prediction accuracy if we set a variable's values to random numbers (i.e. artificially strip it of its predictive power). Based on the graph above, we can see that a few variables dominate other variables in terms of relative importance. Variables such as **sp**(sales to market capitalization ratio), **bm**(book to market ratio), **mvel1**(market capitalization) are extremely important. This result conforms to existing works in the field. For example, multiple works have pointed out that book-to-market ratio has a very strong influence on stock return. Stocks with high book-to-market ratios, also known as value stocks, offers a higher return than stocks with lower book-to-market ratios.

Conclusion and future work

Summary of the results

The result shows that for our particular dataset, Principal Component Regression is the best explanatory model in terms of R squared and Mean Squared Error, with GBRT and ANN coming as the second and third place. However, we expect the performance of ANN to significantly increase if we have a larger dataset.

Lesson learned

Based on the results, we verify that data mining techniques can be applied to empirical equity pricing with success. For our dataset, the Principal Component Analysis seems to work better in terms of R-squared and Mean Square Error.

There are two distinct advantages to utilizing data mining techniques in asset pricing. The first benefit is its ability to reduce dimensions. One of the challenges of asset pricing is to deal with the large amount of variables, predictors and their interactions. Normal regressions tend not to work well in this case, even with a large amount of data. The second advantage is that algorithms such as Artificial Neural Network or Regression Trees are model-independent. I do not need any prior assumption or domain knowledge about pricing theories to conduct research.

If we delve deep into our findings, we can see some interesting findings. For example, momentum factors have demonstrated large explanatory power across all algorithms. This seems to validate finance industry's recent frenzy over using momentum as a gold standard for investment decisions.

Future work

The first aspect that we can continue working on is integrating more data points and predictor variables. For this project, we use roughly 25 variables for 20 large public industrial companies, going all the way back to the 1960s. It is expected that we have a more robust result if we can include more companies and information into our analysis, especially for ANN.

The second aspect is better parameterization. Examples would be number of stages for GBRT, numbers of hidden layers for ANN etc. While we have used optimization algorithms to select for the best parameters, we understand that there are better algorithms out there somewhere. And the calculation capacities of our hardware also limit us to the number of iterations and number of optimization attempts we can make.

The third aspect is that we can use more Bayesian statistics into our analysis. For example, instead of using frequentist regression in our PCR, we can do a Monte Carlo Markov Chain(MCMC) panel regression. Or instead of using frequentist regression as the activation function in ANN, we can use MCMC regression. Another aspect where we can incorporate more Bayesian statistic is model comparison. In addition to R squared or MSE, we can use log marginal likelihood or predictive likelihood.

The fourth thing is that, based on our findings from this project, we can try to build a more complicated and realistic portfolio. What kind of return can our strategy offer? And how would that return change once we account for factors such as tax, transaction cost, short sell limit etc.

Credit distribution

Renyan Zhang: Data Analysis(PCR, ANN) in R, Summary, Conclusion, Future work

Di Zhou:

Wu Yang: discussion, data pre-pocessing, 2 methods, documentation