Machine Learning Engineer Nanodegree

Capstone Project Report

Guanyi Yang

September 27, 2019

I. Domain Background

Looking at various fields of quantitative investment, we find that multi-factor stock selection is the most suitable framework for transforming into machine learning. In particular, massive increases in data, access to low cost computing power, and advances in machine learning have significantly changed the financial industry. This paper will explore how several machine and deep learning algorithms can be applied in China A-share market.

II. Problem Statement

A common method for price prediction are regression-based strategies. Such strategies use regression analysis to extrapolate a trend to derive a financial instruments direction of future price movement. The problem to be solved, then, is understanding the relationship between historical data and future price prediction. The classical multi-factor model is expressed as:

$$\tilde{r} = \sum_{k=1}^{K} x_{ik} * \tilde{f_k} + \mu_i$$

X_{ik}: The factor exposure of the stock i on the factor k

$\tilde{f_k}$: Factor return of factor k

μ_i: The residual yield of stock i

The essence of classical multi-factor model is linear regression. Factor exposure x_{ik} can be observed as an independent variable of linear regression. Stock return--$\tilde{r}$, is equivalent to the dependent variable of linear regression. By fitting the historical data of x_{ik} and $\tilde{r}$, we can calculate the regression coefficient $\tilde{f_k}$, which is the estimator of factor return. Without taking risk into consideration, we can select stocks and allocate their weight only based on $\tilde{r}$ and constraints. For example, by ranking industries, selecting the top N stocks in each industry, allocate weights according to the principle of industry neutrality, and get the portfolio of the next phase.

Support vector machine (SVM) is one of the most widely used machine learning methods. Linear support vector machine can solve linear classification problems, kernel support vector machine is mainly for nonlinear classification problems. In this report, we apply support vector machine to multi-factor stock selection, focusing on the following issues:

1. Firstly, model selection. Is there any improvement in the performance of support vector machines compared with linear regression models? Does the nonlinear classifier such as polynomial kernel, Sigmoid kernel and Gaussian kernel support vector machine, have advantages in classification performance compared with the linear classifier represented by linear support vector machine? Is there any difference between the predictive ability of support vector regression and support vector classifier?
2. Secondly, parameter optimization. SVM depends much more on parameters than generalized linear models. The SVM contains two important parameters: the penalty coefficient C and gamma (γ) value. In the context of the problem of combining multiple factors, what is the most reasonable value of parameters? Which indicators should be used to determine the optimal parameters?
3. Finally, portfolio construction. After measuring the performance of different support vector machine models, how to use the prediction results of the model to construct a strategy combination for back-test? What are the similarities and differences of stock selection effect of each model in A-share pools?

III. Support Vector Machine (SVM) Introduction

Support vector machine (SVM) is one of the most widely used machine learning methods. Linear support vector machine can solve linear classification problems, kernel support vector machine is mainly for nonlinear classification problems.

i. Linear SVM

We are given a training dataset of n points of the form $(\mathrm{x}_1, y_1), \dots, (x_n, y_n)$ where the y_i are either 1 or -1, each indicating the class to which the point $\mathrm{x_i}$ belongs. Each $\mathrm{x_i}$ is a p-dimensional real vector. We want to find the "maximum-margin hyperplane" that divides the group of points $\mathrm{x_i}$ for which $y_i = 1$ from the group of points for which $y_i = -1$, which is defined so that the distance between the hyperplane and the nearest point $\mathrm{x_i}$ from either group is maximized.

Any hyperplane can be written as the set of points x satisfying $\mathrm{w} \cdot \mathrm{x} - \mathrm{b} = 0$ where w is the (not necessarily normalized) normal vector to the hyperplane. This is much like Hesse normal form, except that w is not necessarily a unit vector. The parameter $\frac{\mathrm{b}}{||\mathrm{w}||}$ determines the offset of the hyperplane from the origin along the normal vector w.

ii. Kernel Trick

Suppose now that we would like to learn a nonlinear classification rule which corresponds to a linear classification rule for the transformed data points $\varphi(\mathrm{x_i})$. Moreover, we are given a kernel function k which satisfies $\mathrm{k}(\mathrm{x_i} \cdot \mathrm{x_j}) = \varphi(\mathrm{x_i}) \cdot \varphi(\mathrm{x_j})$. We know the classification vector w in the transformed space satisfies $\mathrm{w} = \sum_{i=1}^{n} c_i y_i \varphi(x_i)$, where, the c_i are obtained by solving the optimization problem maximize $\mathrm{f}(\mathrm{c}_1, \ldots, c_n) = \sum_{i=1}^{n} c_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} c_i y_i \left(\varphi(\mathrm{x_i}) \cdot \varphi(\mathrm{x_j})\right) c_j y_j = \sum_{i=1}^{n} c_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} c_i y_i \mathrm{k}(\mathrm{x_i} \cdot \mathrm{x_j})\, c_j y_j$ subject to $\sum_{i=1}^{n} c_i\, y_i = 0$, and $0 \le c_i \le \frac{1}{2n\lambda}$ for all i.

The coefficients c_i can be solved for using quadratic programming, as before. Again, we can find some index i such that $0 < c_i < \frac{1}{2n\lambda}$, so that $\varphi(\mathrm{x_i})$ lies on the boundary of the margin in the transformed space, and then solve $\mathrm{b} = \mathrm{w} \cdot \varphi(\mathrm{x_i}) - y_i = \left[\sum_{j=1}^{n} c_j y_j \varphi(\mathrm{x_j})\right] - y_i = \left[\sum_{j=1}^{n} c_j y_j \mathrm{k}(\mathrm{x}_j \cdot \mathrm{x_i})\right] - y_i$.

Finally, $2 \rightarrow \mathrm{sgn}(\mathrm{w} \cdot \varphi(z) - \mathrm{b}) = \mathrm{sgn}(\left[c_j y_j \mathrm{k}(\mathrm{x}_i \cdot \mathrm{z})\right] - \mathrm{b})$

iii. Kernel Function in common use

1. Linear kernel: $\mathrm{k}(\mathrm{x_i} \cdot \mathrm{x_j}) =< \mathrm{x_i}, \mathrm{x_j} > \sum_{\mathrm{k}=1}^{\mathrm{p}} \mathrm{x}_i^{(\mathrm{k})} \mathrm{x}_\mathrm{j}^{(\mathrm{k})}$

2. Multi-order polynomial kernel: $\mathrm{k}(\mathrm{x_i} \cdot \mathrm{x_j}) = (\gamma \sum_{\mathrm{k}=1}^{\mathrm{p}} \mathrm{x}_i^{(\mathrm{k})} \mathrm{x}_\mathrm{j}^{(\mathrm{k})} + 1)^{d}$

3. Sigmoid kernel: $\mathrm{k}(\mathrm{x_i} \cdot \mathrm{x_j}) = \tanh(\gamma < \mathrm{x_i}, \mathrm{x_j} > +1) = \tanh(\gamma \sum_{\mathrm{k}=1}^{\mathrm{p}} \left(\mathrm{x}_i^{(\mathrm{k})} \mathrm{x}_\mathrm{j}^{(\mathrm{k})} + 1\right))$

4. Gaussian kernel: $\mathrm{k}(\mathrm{x_i} \cdot \mathrm{x_j}) = \exp(-\gamma(\sum_{\mathrm{k}=1}^{\mathrm{p}} \left(\mathrm{x}_i^{(\mathrm{k})} - \mathrm{x}_\mathrm{j}^{(\mathrm{k})}\right)^2))$

IV. SVM Solution Process

i. Data acquisition

All A-share constituent stocks, excluding. The ST shares, stocks suspended in the next trading day of each cross-section period and the stocks listed within 3 months were excluded. We use data are from 2007 to 2019 and broken into training and test sample.

In sample training data are from Jan 31, 2007 to Dec 31, 2012, which includes 72 months.

Test data are from Jan 31, 2013 to Aug 31, 2019, which includes 80 months.

ii. Feature and label extraction

On the last trading day of each month, factor exposure was calculated as the original feature of the sample. Calculate the excess return of individual stocks for the next month as the sample label.

iii. Eigenvalue preprocessing

1. Median Absolute Deviation (MAD). Let the exposure sequence of a certain factor on all stocks in period T be D_i. D_M is the median of the sequence. D_{M1} is the median of $|D_i - D_M|$. Then, reset all the numbers in sequence D_i that are greater than $D_M + 5D_{M1}$ to $D_M + 5D_{M1}$, and all the numbers in sequence D_i that are less than $D_M - 5D_{M1}$ to $D_M - 5D_{M1}$.
2. Filling missing data. Fill the missing value in the factor exposure sequence with the average value of the first-class industry CITIC after the new factor exposure sequence is obtained.
3. Neutral market value of the industry. Linear regression was made on the dummy variable of the industry and the log market value, and the residual was taken as the new factor exposure.
4. Standardization. To get a new sequence, which approximation obeys the N(0,1) distribution.
5. PCA (principal component analysis). To avoid collinearity between features, we use PCA to get new features after transformation.

iv. Synthesis of training sets and cross validation sets

Classification: At the end of each month, the stocks before and after 30% of next month's earnings were selected as positive and negative examples respectively. The samples of 72 months were merged, and 90% of the samples were randomly selected as the training set and the remaining 10% as the cross-validation set.

v. In sample training

Use SVM to train the training set. Selects five different types of kernel functions: linear kernel, 3-order polynomial kernel, 7-order polynomial kernel, Sigmoid kernel and Gaussian kernel.

vi. Cross validation callbacks

After determining the optimal parameters, the preprocessed characteristics of all samples in the cross section period at the end of T month were taken as the input of the model and the predicted values of T+1 month were obtained, based on which we can build portfolios.

vii. Evaluation Method

Evaluation indicators include two aspects: one is the accuracy of test set, AUG and other indicators to measure the performance of the model; Secondly, the performance of the portfolio strategy constructed in the previous step, such as annualized excess return, information ratio etc.

viii. Parameters and Factors

1. Parameters of SVM models

After data cleaning and preparing, we use SVM or SVR to train the training set. SVM selects five different kernel functions: linear kernel, 3-order polynomial kernel, 7-order polynomial kernel, Sigmoid kernel and Gaussian kernel. The SVR selects Gaussian kernel. Then, after the model training, using this model to predict the cross validation set. Parameters with the highest AUC (SVM) or IC (SVR) in the cross validation set are selected as the optimal parameters of the model. Use these optimal parameters to test out of the samples.

Method	**Kernel Function**	**Kernel Parameters**
SVM	Linear Kernel	C = 3e-4
	3-order polynomial kernel	C=0.1, γ=0.01
	7-order polynomial kernel	C=1e-4, γ=0.003
	Sigmoid kernel	C=10, γ=3e-5
	Gaussian kernel	C=1, γ=0.01
Linear Regression	-	-

2. Factors Selected and descriptions

Classification	**Factors**	**Description**
Technical factor	VOLUME_RATIO_5D	Volume/ Past 5 days average volume
	VAM_1M	MA of VAM volume (a month)
	VAM_5M	MA of VAM volume (5 days)
	VAM_22M	MA of VAM volume (22 days)

	VAM_60M	MA of VAM volume (60 days)
	AMA_1W	MA of AMA amount (a week)
	AMA_1M	MA of AMA amount (a month)
	AMA_1Q	MA of AMA amount (a quarter)
	VMACD	VMACD (DIF: 12, 26, 9 days)
	VMACD_DEA	VMACD (DEA: 12, 26, 9 days)
	VMACD_MACD	VMACD (MACD: 12, 26, 9 days)
	VOSC	Volume Oscillator (12, 26 days)
	TAPI_16D	Weighted index transaction value (16 days)
	TAPI_6D	Weighted index transaction value (6 days)
	VSTD_10D	Volume standard deviation (10 days)
Financial Factors	S_VAL_MV	Total company capitalization
	S_DQ_MV	Total float market capitalization
	S_PQ_HIGH_52W_	The highest price of past 52 weeks
	S_PQ_LOW_52W_	The lowest price of past 52 weeks
	S_VAL_PE	Price / earnings
	S_VAL_PB_NEW	Price / book value
	S_VAL_PE_TTM	Price / earnings (TTM)
	S_VAL_PCF_OCFTTM	Price / operating cash flow (TTM)
	S_VAL_PCF_NCFTTM	Price / Free cash flow (TTM)
	S_VAL_PS_TTM	Price / sales (TTM)

	S_DQ_TURN	Turnover rate
	TOT_SHR_TODAY	Total shares
	FLOAT_A_SHR_TODAY	Total float shares
	S_DQ_CLOSE_TODAY	Close price
	S_PRICE_DIV_DPS	Price / Dividend
	FREE_SHARES_TODAY	Total free float shares
	NET_PROFIT_PARENT_COMP_TTM	Net parent company profit (TTM)
	NET_ASSETS_TODAY	Net assets
	NET_CASH_FLOWS_OPER_ACT_TTM	Net operating cash flow
	OPER_REV_TTM	Operating revenue
	NET_INCR_CASH_CASH_EQU_TTM	Net increasing cash and cash equivalent
	UP_DOWN_LIMIT_STATUS	Status

V. Results of accuracy of test set

The results of accuracy of test set of all models are as following:

Method	Kernel Function	Accuracy	ACU
SVM	Linear Kernel	0.62	0.66
	3-order polynomial kernel	0.53	0.71
	7-order polynomial kernel	0.54	0.60
	Sigmoid kernel	0.61	0.65
	Gaussian kernel	0.64	0.69
Linear Regression	-	-	0.66

VI. Strategies Construction and Performance

Based on the SVM models, selecting the 100 stocks that are most likely to rise each month and allocate assets equally. Then calculate the return and net value of the strategy.

We got annual excess return, annual excess volatility and information ratios of models are as follows: (details and more plots are in html files in attached zip package)

Method	Kernel Function	Excess Return	Excess Volatility	Information Ratio
SVM	Linear Kernel	17%	7%	2.47
	3-order polynomial kernel	9%	6%	1.50
	7-order polynomial kernel	8%	5%	1.75
	Sigmoid kernel	13%	7%	1.88
	Gaussian kernel	16%	6%	2.56
Linear Regression	-	16%	7%	2.31

VII. Kernel Function Comparison of SVM

We compare five different kernel functions: gaussian kernel, linear kernel, third-order polynomial kernel, seventh-order polynomial kernel and Sigmoid kernel. The five methods all contain the penalty coefficient C; In addition, Gaussian, polynomial, and Sigmoid nuclei also contain gamma values. Finally, we select the final selected parameters C and γ.

As the above results, the SVM model with linear kernel and Gaussian kernel have the highest information ratio, which means that they have higher ability to obtain excess returns over the market level under a given risk.

In conclusion, the performance of SVM models with different kernel functions is quite different. From the perspective of excess return, Gaussian kernel SVM performs best, and the second one is linear kernel. The 3-order polynomial kernel performs the worst.

VIII. Prospect and Future Improvement

The penalty coefficient C and gamma are the most important parameters of SVM model. In the future, we can try to find the global optimal solution of C and gamma values at the same time by using grid search. Then, we can choose the best parameters of C and gamma of each models, which would definitely improve SVM models of all kinds of kernel functions.

On the other hand, we can apply SVM models to select stocks in different index, industries etc. to construct new portfolios with different weights or to classify the portfolios using layered back-test compared with benchmark model and index. In order to achieve this method, we need to clean the data using industry neutralization when we prepared data, then, all stocks are sorted by factor size within each first-level industry, and each industry is divided into N stratified portfolios. The evaluation metrics are same as what we used in this report.