Feature Scaling for Financial Machine Learning

Jieli Shen*
jlshen2011@gmail.com

This version: May 15, 2022

Abstract

Machine learning have made a large number of novel applications in various domains in finance. Though more complex and advanced models have been proposed and explored in literatures, the input data to the models and how the raw features are preprocessed, remains important. This article provides an overview of feature scaling that is commonly performed during data preprocessing in financial machine learning applications.

1 Introduction

Machine learning has made a large number of novel applications in various domains and finance is of no exception. See for example, [5], [8], [4], [10] for some recent books on financial machine learnings. The journey of building a financial machine learning model starts with collecting proper data. The next effort goes into preprocessing the raw data to extract the features as inputs to the model. During data preprocessing, feature scaling refers to a class of methods used to normalize the range of features, and are important for building successful financial machine learning applications.

First, feature scaling is required by some machine learning models to guarantee they work as expected. For example, l_1 and l_2 regularized methods (e.g., Lasso and ridge regressions, support vector machines) assume all features are centered around zero and have variance of the same order. If a feature has variance that is orders of magnitude larger than others, it might dominate the objective function and make the estimator unable to learn from other features correctly. Feature scaling is also advised in other methods for better training speed and performance. It is a common practice to standardize features before training a neural network since gradient descent converges much faster with feature scaling than without it [9]. Indeed, financial data come from a large variety of scales:

*The article is independently completed by the author and does not reflect the views of the author's current or past affiliations.

"price returns are most of the time smaller than one in absolute value; stock volatility lies usually between 5% and 80%; market capitalization is expressed in million or billion units of a particular currency; accounting ratios can have inhomogeneous units." [4]

Second, feature scaling can add additional information compared to raw features that can substantially boost machine learning model performance. As mentioned in [16], "a raw factor such as inventory turnover has little meaning unless we compare it to inventory turnovers of other companies, or the company's own historical average." By standardizing the raw turnover with its moving average mean and standard deviation, we can see how today's turnover deviates from its past pattern. For another example, many technical analysis indicators have a bounded range and can be viewed as raw market data scalers in a broader sense. As hand-crafted features, they add values upon the raw market data to the machine learning model.

Last but not the least, some feature scaling methods naturally deal with outliers. Outliers are observations that deviate markedly from the majority of the data, and have been a persistent pain point in empirical financial studies. They can spoil and mislead the training process resulting in longer training time, less accurate models and ultimately poorer results. Some feature scaling methods such as quantile transform are robust to outliers.

This article provides an overview of feature scaling methods applied to financial market data, specially, common feature scaling methods in section 2, feature scaling applied across different companies in section 3.1 and over a company's own history in section 3.2. Financial examples are provided whenever necessary. Note that there are many other alternative raw data sources. [11] categorized them into text data, macroeconomics data, knowledge graph data, image data, fundamental data, analytics data in addition to market data. Market data, however, is most important and popular data source because: 1) It can be used to extract rich features (such as raw OHLCV or technical analysis indicators) but is required for label construction in most of the financial machine learning applications. 2) It is easily accessible with large data volume compared to other sources. 3) Efficient market hypothesis claims that stock prices already reflect all available information.

2 Scaling Methods

2.1 Standardization

Standardization transforms a feature by subtracting the population mean μ from an individual raw feature and then dividing the difference by the population standard deviation. σ

$$z_i = \frac{x_i - \mu}{\sigma}.$$

Standard scores are most commonly called z-scores in statistics.

Computing a z-score requires knowledge of the mean and standard deviation of the complete population to which a data point belongs. In practice, we

only have a sample of observations from the population, then the analogous computation using the sample mean $\hat{\mu}$ and sample standard deviation $\hat{s}$ yields the t-statistic (of $n-1$ degrees of freedom where n is the sample size):

$$z_i = \frac{x_i - \hat{\mu}}{s}.$$

As n increase, the t-statistic and z-score become the same in probability because of the law of large numbers.

2.2 Min-max Scaling

An alternative to standardization is scaling features to lie between a given minimum and maximum value, often between zero and one so that the maximum absolute value of each feature is scaled to unit size:

$$z_i = \frac{x_i - x_{\min}}{x_{\max} - x_{\min}}.$$

It is also feasible to scale the feature values to arbitrarily fixed range $[a, b]$ through a further linear transform:

$$z_i = a + (b-a) \cdot \frac{x_i - x_{\min}}{x_{\max} - x_{\min}}.$$

2.3 Trimming and Winsorizing

Estimation of the min, max, and moments may be heavily influenced by outliers and potential data errors. Trimming and winsorizing are two ways to mitigate the effect of extreme outliers. Trimming excludes the outlier values from the estimation and the scaled scores of these outliers are simply set zero. winsorizing caps outliers to specified values and uses the capped values in estimation. The thresholds for trimming or winsorizing are often based on quantiles of the raw features.

The choice to winsorize, trim or retain the outliers depends on individual case. For example, market cap is a commonly used feature that measures the size of a company. Stocks with extreme market cap in the its top or bottom quantiles tend to behave similarly, thus it's better to winsorize or simply retain these values as they contains meaningful information of price movement pattern. An opposite example is earning yield [16]. Extremely high historical earnings yield may from one-time sale of a valuable division and may not reflect the company's future earnings ability. In this case, trimming outliers and assigning zero scores to these stocks can be a fair treatment.

2.4 Quantile Transform

Quantile transforms put all features into the same desired distribution based on the formula $G^{-1}(F(x))$ where F is the cumulative distribution function of

the feature and G^{-1} is the inverse probability function of the desired output distribution. This formula is based on the facts that 1) if X is a random variable with a continuous cumulative distribution function F then $F(X) \sim \text{Uniform}[0,1]$; 2) if $U \sim \text{Uniform}[0,1]$ then $Z := G^{-1}(U)$ is distributed with cumulative distribution function G.

Usually F is unknow and is estimated by empirical distribution function: $\hat{F}(x) = \frac{\text{number of } x_i \leq x}{n}$. As a special of case where $G(u) = u, \forall u \in [0,1]$, $G^{-1}(\hat{F}(x))$ transforms x_i to its normalized rank.

Similar to trimming and winsorizing, quantile transform is also suitable when there are extreme values and outliers in the raw feature. Consider market cap again and the universe of all stocks in Rusell3000. The market cap distribution is right skewed because of a few mega-cap stocks. Standardizing market cap for all the stocks in Russell 3000 results z-scores of little value [16] while quantile transform smooths out unusual distributions and is less influenced by outliers. The drawback is that some information is lost during the transform, and as is the issue with nonlinear transforms in general, correlations and distances within and across features are distorted. If the scales of the raw feature have information beyond the ranks, then the ranks could be less efficient with less predictive power. The choice between trimming/winsorizing and quantile transform is therefore a tradeoff between robustness and statistical efficiency.

2.5 Power Transform

Normality of the features is a often desirable property in many modeling scenarios, for example, Gaussian naive Bayes classifier, graphical lasso [7]. Power transforms are a family of parametric, monotonic transformations that aim to map data from any distribution to as close to a Gaussian distribution as possible. Besides normality, It also helps stabilize variance of the data - a prerequisite for stationarity that is the assumption in many time series statistical models.

One well-known example of power transforms is the Box-Cox transform [2]:

$$z_i = \begin{cases} \dfrac{x_i^\lambda - 1}{\lambda} & \text{if } \lambda \neq 0, \\ \ln x_i & \text{if } \lambda = 0, \end{cases}.$$

The transform is parameterized by λ which is usually determined through maximum likelihood estimation. See [14] for a recent study of how Box-Cox transformation helps forecasting macroeconomic time series.

2.6 Techincal Analysis Indicators

A technical analysis indicator is a mathematical calculation based on historic price and volumes that aims to forecast financial market direction. See, for example, [12], [3], for comprehensive references. They have been widely used hand-crafted features in financial machine learning applications. Some technical indicators are mathematically bounded by their construction. For example, the

relative strength index (RSI), a measure of overbought or oversold levels, has a reading from 0 to 100; and Williams %R, a similar measure, ranges from -100 to 0. Some other indicators, though not strictly bounded, are defined as a ratio between two quantities of the same order of magnitude. For example, the percentage price oscillator (PPO) calculates the difference between the 12-period EMA (exponential moving average) of prices and the 26-period EMA and normalize it by the 26-period EMA; and the Stochastic %K compares the price difference between most recent close and the 14-period low to the difference between 14-period high and 14-period low. In a broader sense, these technical analysis indicator can be viewed as a special case of scalers of the raw price and volume features.

3 Scaling Axis

3.1 Scaling Across Stocks

In this section we discuss how the feature scaling methods are applied onto the two different axes of the market data - stock and time. Let's begin with the stock axis.

The needs of scaling features across different stocks come from cross-sectional analysis. The seminal Fama-French three-factor model [6] proposed to explain the cross-sectional structure of stock returns by three factors: beta (market portfolio), size (market capitalization), and value (price book-value ratio). Many modern financial machine learning applications often study the joint behavior of different stocks as well to exploit the interrelationships among them; c.f. [1], [13], [15], etc.

Consider standardization. Scaling can be performed by removing the cross-sectional mean of all stocks in the investment universe and dividing the result by the cross-sectional standard deviation at a given time. The process can be repeated for different time point but the mean and standard deviation should be re-estimated each time.

Instead of scaling across all stocks in the investment universe, we can also first divivde the stocks into groups and scale features across stocks within each group. This is similar to what is called neutralization in factor investment. A popular choice of grouping is by industry, because business models, profit drivers and growth potentials often vary by industry, so does the stock price movement pattern. The Global Industry Classification Standard (GICS) categories all major U.S. public companies into sectors, industry groups, industries and sub-industries with increasing level level of granularities, and serves as standard industry grouping criterions. Alternatives include North American Industry Classification System (NAICS), Bloomberg Industry Classification Systems (BICS), etc. Another choice of grouping is by geolocation such as by country or global markets (North America, Europe, EMEA), Asia Pacific, etc.), when it comes to cross-country investment.

[16] described the general principle of deciding the grouping criterion is that

"we should choose the most parsimonious model unless the added complexity of neutralization has clear rationale and yields significant better risk-adjusted returns." This is in spirit the Occam's razor problem-solving principle that "entities should not be multiplied beyond necessity" and the bias-variance tradeoff in statistical learning theory. With the same amount of stocks, the more groups there are, the better we are able to fit individual stocks but at the risk of more volatile estimation because each group is allocated fewer number of stocks.

Thus far, we have assigned equal weight to each stock. Equal weights ignore market values of the stocks; a small-cap $300 million stock is regarded the same as a mega-cap $300 billion stock. In practice, investment professionals generally think that large-cap stocks better represent their corresponding industries or the whole investment universe because of larger market size, trading volume, and liquidity, and should be assigned higher weights when calculating the means and standard deviations. Weighting schemes include weights proportional to the market caps and square roots of them. In the former scheme, the average may be dominated by a few mega-cap stocks and the issue is alleviated in the latter scheme.

3.2 Scaling Across Time

In contrast to cross-sectional scaling, time series scaling are used to compare a feature with its own historical value. Consider standardization across time. The mean is the moving average of historical raw feature values of the same stock, and similar for the standard deviation. The result time series z-score can be used to compare, for example, a stock's trading volume compared to its past average, and the extent of deviation measured in multiples of standard deviation.

We can apply the idea of assigning unequal weights in stock axis described in section 3.1 into the time domain as well. For example, a weighted moving average (WMA) in technical analysis assigns weights that decrease in arithmetical progression. In an k-period WMA the latest day has weight k, the second latest $k-1$, etc., down to one:

$$\mathrm{WMA}_{t,k} = \frac{kx_t + (k-1)x_{t-1} + \cdots + x_{((t-k)+1)}}{k + (k-1) + \cdots + 1}.$$

An exponential moving average (EMA) applies weighting factors which decrease exponentially. EMA can be calculated recursively as

$$\mathrm{EMA}_t = \alpha x_t + (1-\alpha)\mathrm{EMA}_{t-1}.$$

There are many other time-series weighting schemes such as volume weight that weights each time period in proportion to its trading volume, moving median that also estimate the underlying trend but is robust to rapid shocks or other anomalies, etc. The choice of weighting schemes, together with any hyperparameters therein (such as lookback window length), is more of an art than a science. A good practical is to try out different candidates and choose the

one(s) that yields best risk-adjusted returns on validation/test data, and discard the rest since these features are usually highly correlated with each other.

It's worth mentioning that for prediction related applications, no information beyond the time point at which the time-series scaling is performed should be used to avoid the look-ahead bias. Otherwise not only does it leak future information that can to artificially overestimated performance, it's also impossible to make decisions based on features that is not even available yet in real trading/investing.

References

[1] Abe, M., and Nakayama, H. Deep learning for forecasting stock returns in the cross-section. *arXiv:1801.01777v4* (2018).

[2] Box, G. E. P., and Cox, D. R. An analysis of transformations. *Journal of the Royal Statistical Society, Series B 26*, 2 (1964), 211–252.

[3] Colby, R. W. *The Encyclopedia Of Technical Market Indicators*, 2nd ed. McGraw-Hill Education, 2002.

[4] Coqueret, G., and Guida, T. *Machine Learning for Factor Investing: R Version*. Chapman and Hall/CRC, 2020.

[5] de Prado, M. L. *Advances in Financial Machine Learning*. Wiley, 2018.

[6] Fama, E. F., and French, K. R. The cross-section of expected stock returns. *The Journal of Finance 47*, 2 (1992), 427–465.

[7] Friedman, J., Hastie, T., and Tibshirani, R. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics 9*, 3 (2008), 432–441.

[8] Guida, T. *Big Data and Machine Learning in Quantitative Investment*. Wiley, 2019.

[9] Ioffe, S., and Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv:1502.03167* (2015).

[10] Jansen, S. *Machine Learning for Algorithmic Trading*, 2nd ed. Packt, 2020.

[11] Jiang, W. Applications of deep learning in stock market prediction: Recent progress. *Expert Systems with Applications 184* (2021), 115537.

[12] Murphy, J. J. *Technical Analysis of the Financial Markets: A Comprehensive Guide to Trading Methods and Applications*. Prentice Hall Press, 1999.

[13] Rasekhschaffe, K. C., and Jones, R. C. Machine learning for stock selection. *Financial Analysts Journal 75*, 3 (2019), 70–88.

[14] Tommaso, P., and Helmut, L. Does the box-cox transformation help in forecasting macroeconomic time series? *Munich Personal RePEc Archive* (2011).

[15] Wu, W., Chen, J., Yang, Z., and Tindall, M. L. A cross-sectional machine learning approach for hedge fund return prediction and selection. *Management Science 67*, 7 (2021), 3985–4642.

[16] Zhou, X., and Jain, S. *Active Equity Management*. 2014.