Risk Parity Revisited: Learning Ensembles of Stochastic Portfolio Selection Models for Quarterly Asset Rebalancing

Brett D. Whitford
Department of Electrical and Computer Engineering
Ohio State University
Columbus, OH 43210
whitford.15@osu.edu

Noah H. Bayindirli
Department of Electrical and Computer Engineering
Ohio State University
Columbus, OH 43210
bayindirli.1@osu.edu

Abstract

We consider the problem of developing a flexible framework for analyzing portfolio behavior and equity market structure. This is important in a variety of problem settings where investment management practitioners wish to learn an optimal portfolio selection strategy. We describe a method for improving the predictive force of stochastic dominance models by introducing a meta-level to the estimation problem. Namely, we show that a neural network can be trained to identify the optimal portfolio selection method for a given quarter, provided descriptors of a money-market's economic condition and the trailing performance of all strategies under consideration. We show our machine learning approach can ensemble portfolio selection models in such a way that existing individual strategies are outperformed while assuming less risk.

1 Introduction

Portfolio selection aims to mathematically analyze stock market structure and portfolio behavior for maximization of profit and minimization of risk. Various stochastic models have been used over the years, and recently second-order stochastic dominance models have provided promising results.

In this paper, we work with the portfolio selection method known as Stochastic Portfolio Theory (SPT), first introduced and developed by Robert Fernholz [1]. The classical method for comparison is Modern Portfolio Selection (MPT), a normative model proposed by Harry Markowitz [2]. Both methods work to optimize portfolio selection. However, SPT is descriptive instead of normative, making it consistent with the observed behavior of actual markets.

The key issue with the portfolio optimization problem is the definition of preference structure among feasible portfolios [3]. In the Markowitz model, the risk measure of the portfolio is the variance of the return, while other strategies attempt to define a concave and nondecreasing utility function representing the preferences of a risk-averse decision maker. In contrast, SPT looks to a point-wise comparison of some performance functions constructed from their distribution functions [3].

CSE 5523 Machine Learning and Advanced Pattern Recognition, Final Project, April 2017

We propose the novel idea of introducing a meta-level to the portfolio selection problem. Namely, we look to select the most appropriate model to rebalance with at the end of each quarter by incorporating features among multiple portfolio selection models. This framework is achieved with a neural network trained on key risk and performance attributes of trailing quarters.

2 Related Work

Neural networks have recently been achieving state of the art results in many different fields. This can be attributed to the fact that neural networks with one hidden layer and a nonlinear activation function are theoretically able to approximate any function with arbitrary precision. However, neural networks are extremely sensitive to input data, tuning of hyperparameters, and initialization. If a representation of a learner is chosen that does not include the optimal classifier in the feature space, there is no hope of learning the optimal classifier regardless of how long the network is trained.

With these considerations in mind, we turn to the most recent advantages in portfolio theory to aid our network in learning an optimal trading strategy from financial data. We primarily concentrate on a particular portfolio construction framework known as Stochastic Portfolio Theory [1]. In SPT, the amount by which a portfolio $\pi(\cdot)$ outperforms the vector process of market weights $\mu(\cdot)$ is irrelevant as long as the portfolio strongly outperforms the market over a time-horizon $[0, T]$. This is known as a relative arbitrage and is show in Equation 1, where V^π is a wealth process of an investor implementing portfolio $\pi(\cdot)$.

$$\mathbb{P}((V^\pi(T)) > V^\mu(T)) = 1 \tag{1}$$

This class of portfolios is known as functionally generated portfolios and their performance against a benchmark follows “Fernholz’ Master Equation”, shown in Equation 2.

$$\log\left(\frac{V^\pi(T)}{V^\mu(T)}\right) = \log\left(\frac{\mathbf{G}(\mu(T))}{\mathbf{G}(\mu(0))}\right) + \int_0^T \mathfrak{g}(t)dt \tag{2}$$

where $\mathbf{G}$ is the generating function of a functionally-generated portfolio and

$$\mathfrak{g}(t) := -\sum_{i,j=1}^{n} \frac{D^2_{ij}\mathbf{G}\left(\mu(t)\right)}{2\mathbf{G}\left(\mu(t)\right)}\mu_i(t)\mu_j(t)\tau^\mu_{ij}(t) \tag{3}$$

is the drift process of portfolio $\pi(\cdot)$. The relative covariances are represented by τ^μ_{ij} in Equation 4, which denotes by e_i the i^{th} unit vector in R^n, defined for $1 \le i, j \le n$..

$$\tau^\mu_{ij}(t) := \left(\mu(t) - e_i\right)^T \sigma(t)\sigma^T(t)(\mu(t) - e_j) \tag{4}$$

Recently, many different variations on SPT have been developed. Cumulative Zero-order Epsilon Stochastic Dominance [4] , Roman-Mitra-Zviarovich Second-Order Stochastic Dominance [5], Lizyayev-Ruszczynski approximate Second-Order Stochastic Dominance [6], Luedtke Second-Order Stochastic Dominance [7], Post-Kopa Second-Order Stochastic Dominance [8] have all been proven to consistently outperform traditional index trackers and be robust to the scenario of little or no rebalancing. These six strategies form the basis for our work, while the machine learning perspective of [9] prompted us to assume of challenge of approaching the dataset of [10] in a statistical manner.

3 Proposed Methods

The three most critical parts of a machine learning framework are the representation, evaluation function (objective function), and optimization method. The following section describes our implementation of these three components.

3.1 Model Representation

We build our portfolio selection model around neural networks, as there is an interesting connection between neural networks and Gaussian processes. [11] shows that with certain types of neural networks of one hidden layer, increasing the hidden units to infinity yields a model identical to a Gaussian process with a specific type of covariance function. Therefore, it logically follows that a neural network serves as an appropriate representation for utilizing these methods built around Gaussian processes.

We used MATLAB's Neural Network Toolbox for training, testing, and optimization of our network.

3.1.1 Description of Data

Bruni et al. [10] provided a holistic dataset for our project. The entirety of this data set consisted of several datasets for portfolio selection generated using real-world price values from several major stock markets. The datasets contained weekly return values, adjusted for dividends and for stock splits as well as market index returns. Additionally, the portfolio selection models from [2, 4, 5, 6, 7, 8] were applied to the dataset and the resulting returns and portfolio weights were included.

In particular to this project, we decide to narrow the scope of our work to the S&P 500 as seen in Table 1. For each dataset and for each model on this index, the solutions were computed using a rolling in-sample window of 52 returns observations. Given the in-sample window on the first 52 time periods, we select the portfolio by solving the model, and we evaluate the performance of the selected portfolio on the following 12 (out-of-sample) periods. Next, the in-sample window was updated with the inclusion of the previous 12 out-of-sample periods and the exclusion of the first 12 periods of the previous in-sample window. We then rebalance the portfolio by solving the model again, and this process was repeated to the end of the dataset.

Table 1: Market Data

Market	Assets	Weeks (N)	Time Interval	Rebalances
S&P 500	442	595	Nov 2004-Apr 2016	46

3.1.2 Feature Engineering

The five features utilized as predictor variables were beta, alpha, the standard deviation, the Sharpe Ratio, and R-squared. These features were collected iteratively over financial quarters (loosely taken to be 12 weeks periods).

Beta, β, is observed by the difference in the covariance of the select model X_j and the variance of the market's index return over the 3 year period I at the specified financial quarter n. This calculation is seen in Equation 5.

$$\beta_n = \frac{Cov\left(X_{nj}, I_n\right)}{Var\left(I_n\right)} \tag{5}$$

Alpha, α, is determined by the relationship of β and R_n, the risk-free return at the current period. This implementation may be found in Equation 6.

$$\alpha_n = I_n - \left[R_n + \beta_n \cdot \left(I_n - R_n\right)\right] \tag{6}$$

Equation 7 describes the standard deviation σ of select model X_j given the model's mean μ.

$$\sigma_n = \sqrt{\frac{\sum_{i=1}^{n}(X_{ij} - \mu_n)^2}{n-1}} \tag{7}$$

The Sharpe ratio, S, is described in Equation 8 and is the ratio of the difference in model return X_j and risk-free return and the model's standard deviation.

$$S_n = \frac{X_{nj} - R_n}{\sigma_{nj}} \tag{8}$$

R-squared, r^2, is represented by maximizing the function as seen in Equation 9, where μ_n is the index return's mean.

$$r_n^2 = max\left(0, 1 - \frac{\sum_{i=1}^{n} \left(I_i - X_{ij}\right)^2}{\sum_{i=1}^{n} \left(I_i - \mu_n\right)^2}\right) \tag{9}$$

The response feature $\hat{y_j}$ of the model j is controlled by selecting ψ, the targeted weekly alpha. The intuition behind this feature is is that ψ allows an investment management practitioner to tune their portfolio selection strategy to their desired level of risk-averseness. This function is described by Equation 10.

$$\hat{y_j} = \begin{cases} 0 & \alpha_{n+1} < \psi \\ \alpha_{n+1} & \alpha_{n+1} \geq \psi \end{cases} \tag{10}$$

3.2 Model Evaluation

For evaluation of our model, we computed yearly returns and assumed risk across the test data set. Model return was calculated by cumulating all returns of the selected models at each time window. Model risk was measured by averaging the risk of all chosen strategy, weighted by the number of times each strategy was chosen.

3.3 Model Optimization

MATLAB's Neural Network Toolbox's cross entropy function was used to optimize network performance. This function allowed us to heavily penalize network outputs that were extremely inaccurate while giving very little penalty to those near a correct classification.

4 Experimental Evaluation

4.1 Test Results

The first 31 windows of S&P 500 market data were used for training and the final 14 windows were used for testing the model. Figure 1 shows that our model can outperform all individual strategies while assuming approximately 75% less risk than Post-Kopa Second-Order Stochastic Dominance [8], the highest performing strategy in the test set.

The performance of our strategies does not always outperform every individual strategy. This can be attributed to overconfident posterior predictions. This proved to be a difficult problem to overcome due the nature of our training. As the user pushes $\hat{y}$ closer to a one-hot vector by increasing ψ, the regression problem becomes challenging as the predictor features do not see a similar distinction.

However, even in the worst test scenario where we saw a return of 12.9% per year with risk 0.0327, our model return is comparable the benchmark index which had a test return of 10.8% per year with risk 0.0271. This case is shown in Figure 2.

4.2 Training and Network Architecture

The neural network was trained using cross-entropy as the validation performance metric. Figure 3 shows a graph of validation performance vs training epochs. This allowed us to heavily penalize network outputs that were extremely inaccurate while giving very little penalty to those near a correct classification.

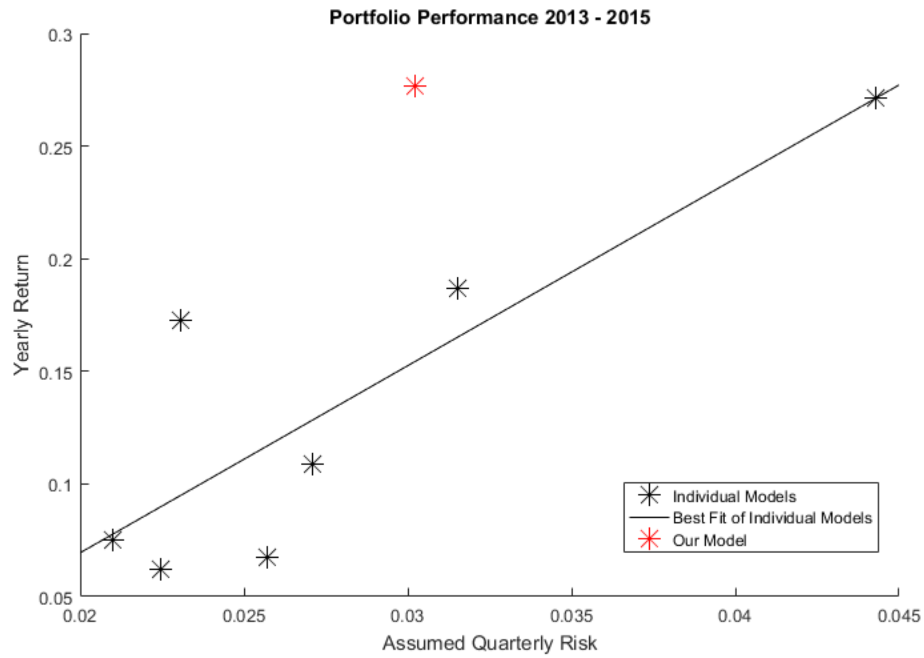

Figure 1: Test results of portfolio performance from 2013 - 2015

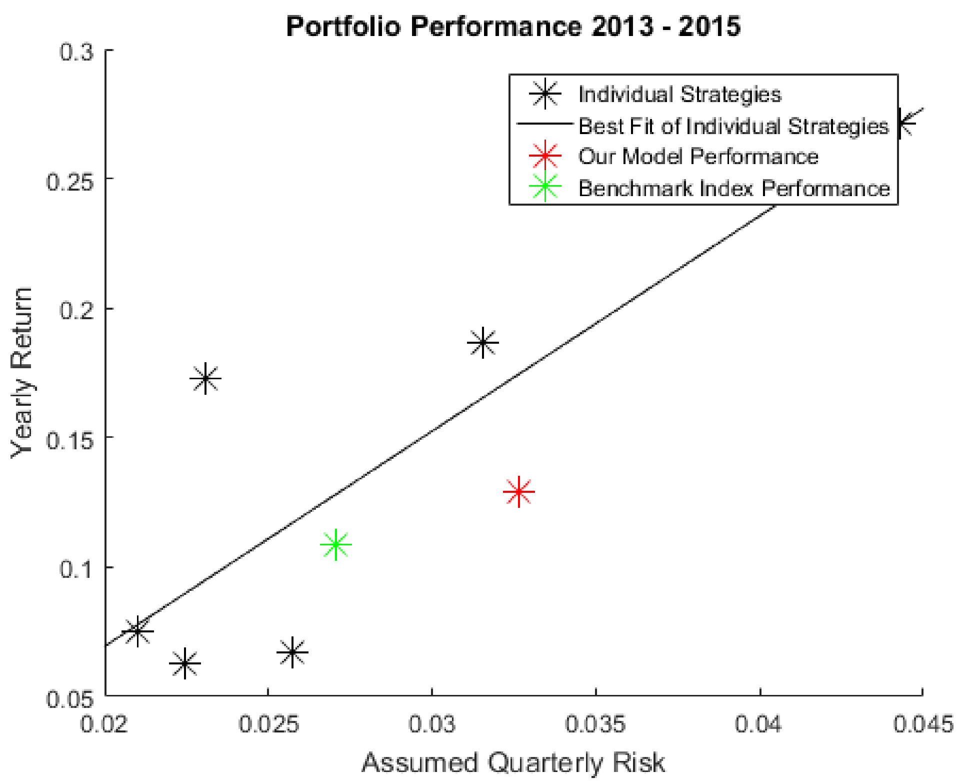

Figure 2: Strong out performance of benchmark even in worst case of training

Our model's performance was best with a high number of nodes in the hidden layer of the network. Over a series of 100 trials, we saw that a network with 50 hidden nodes had the best performance on the test data.

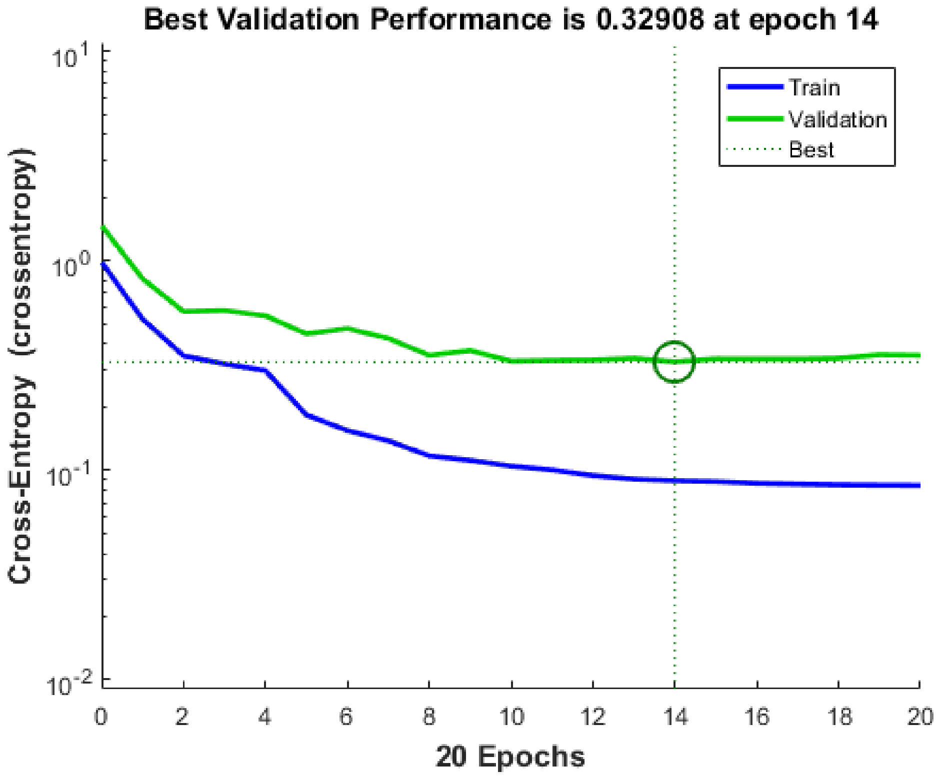

Figure 3: Validation Performance

4.3 Reproducibility

We have implemented the proposed methods in the excellent MATLAB computing environment. We share our source code, data set, and trained models at

`https://github.com/brett-whitford/risk-parity-revisited`

5 Conclusions and Future Work

The inverse problem of stochastic portfolio theory is, given a user-defined portfolio selection criterion, how does one go about constructing suitable investment strategies that meet the desired investment objective? [9] We answer this question through the use of a neural network to choose an appropriate portfolio selection model for the given financial quarter based on specified financial features. Extensive experiments have provided promising results for ensembling current strategies. Results have shown our strategy is able to outperform others by a significant margin while maintaining low risk.

5.1 Future Work

Further extending our work, advancements in cross-evaluation methods across our model will prove to increase test performance of our model. Expansion of our dataset in market figures, model returns, and alternate portfolio selection strategies will also provide for a more in-depth analysis.

Interestingly, [12] has also sparked interest in incorporating Bayesian parameter estimation into our model. This will help the restriction of a smaller dataset, allowing more conclusions to be drawn from the available pool of features and an increase in posterior predictive density accuracy.

Acknowledgments

The authors would like to thank our professor Dr. Alan Ritter for his lessons throughout the semester, as well as Bruni et. al for their data set [10].

A special thank you to Kevin Murphy for his textbook that was used to teach the class and the utility it proved throughout our endeavors [13].

References

[1] E. Fernholz, *Stochastic Portfolio Theory*. Stochastic Modelling and Applied Probability, Springer New York, 2002.

[2] H. Markowitz, "Portfolio selection," *The Journal of Finance*, vol. 7, no. 1, pp. 77–91, 1952.

[3] D. Dentcheva and A. Ruszczyinski, "Portfolio optimization with stochastic dominance constraints," *Society for Industrial and Applied Mathematics*, 2003.

[4] R. Bruni, F. Cesarone, A. Scozzari, and F. Tardella, "Real-world datasets for portfolio selection and solutions of some stochastic dominance portfolio models," *Data in Brief*, vol. 8, pp. 858 – 862, 2016.

[5] D. Roman, G. Mitra, and V. Zverovich, "Enhanced indexation based on second-order stochastic dominance," *European Journal of Operational Research*, vol. 228, no. 1, pp. 273 – 281, 2013.

[6] A. Lizyayev and A. Ruszczyński, "Tractable almost stochastic dominance," *European Journal of Operational Research*, vol. 218, no. 2, pp. 448–455, 2012.

[7] J. Luedtke, "New formulations for optimization under stochastic dominance constraints," *SIAM Journal on Optimization*, vol. 19, no. 3, pp. 1433–1450, 2008.

[8] T. Post and M. Kopa, "General linear formulations of stochastic dominance criteria," *European Journal of Operational Research*, vol. 230, no. 2, pp. 321–332, 2013.

[9] Y.-L. Kom Samo and A. Vervuurt, "Stochastic Portfolio Theory: A Machine Learning Perspective," *Uncertainty in Artificial Intelligence (UAI)*, 2016. UAI 2016.

[10] R. Bruni, F. Cesarone, A. Scozzari, and F. Tardella, "Real-world datasets for portfolio selection and solutions of some stochastic dominance portfolio models," *Data in Brief*, vol. 8, pp. 858 – 862, 2016.

[11] R. M. Neal, *Bayesian Learning for Neural Networks*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 1996.

[12] A. K. Balan, V. Rathod, K. Murphy, and M. Welling, "Bayesian dark knowledge," *CoRR*, vol. abs/1506.04416, 2015.

[13] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012.