Optimization-Based Data Analysis Final Project

Serena Yuan

Finding a solution to optimization affected by parameter uncertainty is a well-studied problem. Robust Optimization provides an approach to optimization under uncertainty, where the uncertainty model is not stochastic, but deterministic and set-based. A solution that is feasible for any realization of the uncertainty in a given set is constructed in a geometrical and tractable fashion.

Uncertainty sets in robust optimization should specify most or all of the possible realizations from the input, typically corresponding to a confidence level under an assumed distribution. Therefore, robust formulations of optimization problems rely very much on uncertainty sets.

In this project we discuss the theory of data-driven construction of uncertainty sets, we implement these sets in two optimization problems in discrete optimization and portfolio optimization, and analyze the best choice of sets.

1 Data-Driven Framework

A big issue in RO includes tractability of several data-driven uncertainty sets as outlined by [2]. It involves the issue of how to structure the uncertainty set U so that the resulting problem is tractable and optimally trades off expected return with loss probability, in the terms of portfolio optimization. We investigate this issue.

1.0.1 Hypothesis Tests and Distributional Uncertainty

A hypothesis test compares two hypotheses, a null-hypothesis H_0 and an alternative hypothesis H_A where each makes a claim about an unknown distribution $\mathbb{P}^*$.

Given a significance level $0 < \delta < 1$ and some data that is drawn from $\mathbb{P}^*$ denoted by X, the test will prescribe a threshold that depends on δ and statistic that depends on X.

If the statistic exceeds the threshold then the hypothesis is rejected, in favor of the alternative.

In the scheme we follow in [2], the key is the use of the confidence region of a statistical hypothesis test to quantify our knowledge about $\mathbb{P}^*$ from the data. We have the given assumption that the data is drawn i.i.d. from an unknown distribution $\mathbb{P}^*$. Then [2] argues that sets that are constructed from the scheme as described in Section 1.1 give a probabilistic guarantee for $\mathbb{P}^*$ at any $\epsilon > 0$.

The interest in hypothesis tests and confidence regions comes from the following observation: when the assumptions of a hypothesis test hold, then the probability, with respect to sampling procedure, that the true distribution of the data $\mathbb{P}^*$ is a member of its confidence region is at least $1 - \delta$. Even without knowing $\mathbb{P}^*$, it is possible to use a hypothesis test to create a set of distributions from the data which will contain $\mathbb{P}^*$ for any specified probability, and therefore they play a major role in designing uncertainty sets. It is possible to substitute the confidence region of a hypothesis test into the possible values for $\mathbb{P}$ to take within our ambiguity about the true distribution $\mathbb{P}^*$ and this relates the support functions of the uncertainty sets with the confidence regions.

More formally, [2] describes the connection between the support function of any set $\mathcal{U}$, denoted as $\phi_{\mathcal{U}}(x) = \max_{u \in \mathcal{U}} u^T x$ and the Value at Risk at level ϵ with respect to x. A risk measure is a central tool in many optimization problems, as it determines in which assets to distribute the total, and the Value-at-Risk is a measure of loss that can be described in a statistical formulation involving hypothesis tests. Given a confidence level $\alpha \in (0, 1)$, the VaR at α is the smallest number ℓ such that the loss L exceeds L with a probability of at most $(1 - \alpha)$. In other words, for L the loss of a portfolio, $VaR_\alpha(L)$ is the level α-quantile. Since support functions of convex sets are convex and positively homogenous, there is also a relationship to the concept of a coherent risk measure ρ; that is, a risk measure that satisfies certain desirable properties such as monotonicity, sub-additivity, homogeneity, and translational invariance. Properties such as sub-additivity and positive homogeneity imply convexity. So in constructing the support function ϕ we define ϕ as the optimal upper bound ρ of the VaR, which has a nominal bound by some ρ, which forms a convex upper bound.

1.1 Data-driven Choice of Uncertainty set procedure

We describe the theory behind the construction of data-driven sets that we will use. As given in [2], the following procedure describes the procedure in finding an ideal uncertainty set. Fix a $0 < \delta < 1$ and $0 < \epsilon < 1$.

1. Let $\mathcal{P}(A)$ be the confidence region of a hypothesis test at level δ for data A.

2. Construct convex, positively homogenous function $\phi(x, A)$ such that

$$\sup_{\mathbb{P}\in\mathcal{P}(A)} VaR_{\epsilon}^{\mathbb{P}}(x) \leq \phi(x, A) \forall x \in \mathbb{R}^d.$$

3. Identify set $\mathcal{U}$ whose support function coincides with ϕ.

This scheme follows from the proposition [[2] Proposition 1] stating that, **Proposition.** A set $\mathcal{U}$ implies a probabilistic guarantee for $\mathbb{P}$ at level ϵ if and only if

$$\phi_{\mathcal{U}} \geq VaR_{\epsilon}^{\mathbb{R}}(x) \forall x \in \mathbb{R}^d. \quad (1)$$

The proposition implies that an ideal set would satisfy $\phi_{\mathcal{U}}(x) = VaR_{\epsilon}^{\mathbb{P}^*}(x) \forall x \in \mathbb{R}^d$, and if such a set $\mathcal{U}$ was found, it would be the smallest (and therefore most efficient) set that implies a probabilistic guarantee, of probability $1-\delta$.

This describes the theoretical procedure in the construction of these sets, and implies certain directions in choosing certain sets. But ultimately, we will use cross-validation to make the best choice of uncertainty sets.

2 Robust Discrete Optimization

We take a problem from Robust Discrete Optimization, and examine how the choice of uncertainty set affects its solution in the robust formulation of the discrete optimization problem.

The binary knapsack problem is:

$$\begin{aligned} \max \quad & \sum_i c_i x_i \\ \text{subject to} \quad & \sum_i w_i x_i \leq q \\ & x \in \{0, 1\}^n. \end{aligned}$$

where q is the knapsack capacity and the weights are uncertain, independently distributed and are from symmetric distributions in $[w_i - \delta, w_i + \delta]$. This interval is the uncertainty set, where at most Γ coefficients can change from nominal value w_i to an arbitrary value in the interval. Therefore, the uncertainty lies in the value of the weights w_i.

Γ is also the level of guarantee with respect to data uncertainty that is desired, which is used in a standard polyhedral uncertainty set formulation of the problem, as $\sum_i z_i \leq \Gamma$. We may assume that $q = 1$ and that each item forms a robust solution independently, $w_i + \delta_i \leq 1 \forall i \in N$.

In our robust formulation of this problem, we replace the weight constraint equation with the constraint that the weight is within the chosen uncertainty set. Therefore, the robust 0/1 knapsack problem is given by:

$$\begin{aligned} \max \ & \sum_i c_i x_i \\ \text{subject to } & \sum_i (w_i + \delta_i z_i) x_i \leq q \ \forall z \in \mathcal{U} \\ & x \in \{0,1\}^n, \end{aligned}$$

for our suitable choice of uncertainty set $\mathcal{U}$.

In our example, we use the normal distribution as the empirical probability distribution $\hat{p}$ that generates the costs and weights data and assume that it has known, finite support within a set of values, denoted by α, where its function is nonzero. This forms the apriori assumption on the distribution. Then we can pair this with a certain hypothesis test. Since it is a discrete distribution, [2] prescribes the use of the χ^2 test to construct the confidence regions and the uncertainty set. We use the $\mathcal{U}^{\chi^2}$.

In the julia code, we will generate size N samples of values within $[0, 1]$ from the normal distribution, and set these to the alphas that form the support of the probability distribution.

In the following table, we list the sizes of costs and weights and the solved optimal value of vector x, given capacity value of 10.

size	$\sum_i c_i x_i$	x
7	1.39667	[0.0,1.0,1.0,0.0,1.0,1.0,1.0]
10	5.03221	[0.0,1.0,1.0,0.0,0.0,0.0,1.0,1.0,1.0,0.0]
15	4.72247	[0.0,1.0,1.0,0.0,0.0,1.0,0.0,1.0,0.0,1.0,0.0,0.0,1.0,1.0,1.0]
20	3.42998	[0.0,1.0,0.0,0.0,1.0,0.0,0.0,0.0,0.0,1.0,1.0,0.0,0.0,0.0,0.0,0.0,0.0,1.0,0.0,0.0]

Table 1: Chi-squared Uncertainty Set optimized values

These values in Table 1 may be compared to the solutions yielded by the polyhedral uncertainty set in Table 2.

size	$\sum_i c_i x_i$	x
7	1.39667	[0.0,1.0,1.0,0.0,1.0,1.0,1.0]
10	5.03221	[0.0,1.0,1.0,0.0,0.0,0.0,1.0,1.0,1.0,0.0]
15	4.72247	[0.0,1.0,1.0,0.0,0.0,1.0,0.0,1.0,0.0,1.0,0.0,0.0,1.0,1.0,1.0]
20	3.42998	[0.0,1.0,0.0,0.0,1.0,0.0,0.0,0.0,0.0,1.0,1.0,0.0,0.0,0.0,0.0,0.0,0.0,1.0,0.0,0.0]

Table 2: Polyhedral Uncertainty Set optimized values

2.0.1 Analysis of results

We may analyze the effectiveness of these solutions by looking at the parameter we wanted to maximize, $\sum_i c_i x_i$.

For all of the sizes, we see that the value $\sum_i c_i x_i$ does not differ, and the choice of uncertainty set does not change the effectiveness of the solution.

3 Time Series Data-based Optimization

There are relevant optimization problems using financial time series, including portfolio selection, asset allocation, and risk management.

The problem we examine is the risk aversion formulation of the mean variance portfolio optimization problem, called the min-max robust mean variance portfolio problem.

Given a time series data set of prices for an asset, $T = \{p_1, \cdots, p_N\}$ of prices at times t_i, the returns are given by $R_i = p_i/(p_{i-1} - 1)$, the expected return is given by $R = \sum_i R_i/N$, and variance by $V = \sum_i (R_i - R)^2/N$. Let $\mu \in \mathbb{R}^n$ be the vector of mean returns $\mu = R/N$ of the n risky assets and Q be the positive semi-definite covariance matrix.

We can construct a portfolio of n assets with portfolio percentage weights x_j whose sum is bounded by 1,

$$x_j, j = 1, \cdots, m, \sum_j x_j = 1$$

and this gives portfolio return $R' = \sum_j x_j * R_j$ and variance V' and mean and variance of the portfolio return are expressed as $(x^T \mu, x^T Q x)$ for $Q = \text{Cov}(r_t)$, where μ is the random vector of mean returns. Now a mean value efficient portfolio solves the problem:

$$\min_x \; -\mu^T x + \lambda x^T Q x \tag{2}$$

$$\text{s.t. } x \in \Omega \tag{3}$$

for each risk aversion parameter, $\lambda \geq 0$ (typically between 1 and 10 [1]) and Ω the feasible portfolio set given by $\Omega = \{x \in \mathbb{R}^n |\ e^T x = 1, x \geq 0\}$ for e the vector of all ones (weight constraint).

Indeed, to choose an optimal portfolio, one can choose $\lambda = 0$ for the maximum-return portfolio, or choose the maximum λ for the minimum-variance portfolio, to avoid risk. The *efficient frontier* is a curve in the space of standard deviation and mean, which outlines the feasible portfolio which outlines possible solutions from the maximum-return to the minimum-variance portfolio. Even when future risks and returns are assumed to be known, there is a lot of uncertainty involving investment decisions; for a correct forecast of 10% return and 20% standard deviation, the range of possible returns is approximately −10% to 30% two-thirds of the time [4].

The min-max robust formulation of the original mean-variance problem is:

$$\min_x \max_{\mu \in S_\mu, Q \in S_Q} -\mu^T x + \lambda x^T Q x \tag{4}$$

$$\text{s.t. } x \in \Omega \tag{5}$$

for uncertainty sets S_μ, S_Q, and its formulation illustrates the interest in solving for the optimal efficient frontier.

For a known covariance matrix Q from data, the uncertain parameter arises in the mean returns of assets μ[3].

So the robust formulation, given the choice of uncertainty set, may be given by,

$$\min_x - \mu^T x + \lambda x^T Q x \ \ \forall \mu \in \mathcal{U}_\mu \tag{6}$$

$$\text{s.t. } x \in \Omega \tag{7}$$

We may further explain the derivation of this robust formulation in the Appendix.

Now we must choose the set $\mathcal{U}_\mu$ for the time series data.

We will implement this as well as the formulations involving data-driven uncertainty sets. The times series data is from the WRDS library[5], the data for the monthly expected returns for stocks from certain companies is given over a period of several years.

3.1 Times Series Experiment

The Ebay dataset consists of monthly assets in 5 categories: *Holding Period Returns, Holding Period Returns without Dividends, Value-Weighted*

Returns (includes distributions), Value-Weighted Returns (excluding dividends), and Equal-Weighted Returns (includes distributions). The dates are from January 2007 to December 2015.

Now we use sets $\mathcal{U}_\epsilon^M, \mathcal{U}_\epsilon^{CS}$. We do not consider sets $\mathcal{U}_\epsilon^I, \mathcal{U}_\epsilon^{FB}$ since we do not know apriori that the returns are independent, and we do not use $\mathcal{U}^{DY}$ since it requires data with larger magnitude than the data we use, which has been adjusted according to the total market. We compare the results with the results given by the polyhedral uncertainty set.

Using the polyhedral uncertainty set yielded results such that the problem was not bounded, and after scaling the return samples by 100 times, it yielded the solution vector, $[0.0, 0.0, 1.0, 0.0, 0.0]$, showing that the set does not handle this optimization problem well in julia.

We set $\alpha, \epsilon \in [10, 30]\%$ for all the sets, and record the results for different parameters.

We have the following optimal portfolios given in Table 3 and Table 4.

(α, ϵ)	$\mathcal{U}^{CS}$ portfolio
(0.2, 0.2)	[0.09914,0.15208,0.27303,0.25735,0.21839]
(0.3, 0.2)	[0.13124,0.16626,0.31077,0.27948,0.11225]
(0.3,0.1)	[0.11567,0.13724,0.32804,0.18252,0.23652]
(0.2,0.155)	[0.15485,0.14312,0.21336,0.32789,0.16079]

Table 3: $\mathcal{U}^{CS}$ Set optimized portfolios

(α, ϵ)	$\mathcal{U}^{M}$ portfolio
(0.2, 0.2)	[1.8637e-7,1.8637e-7,0.9999,4.08738e-7,6.8246e-7]
(0.3, 0.2)	[7.48245e-10,7.48246e-10,0.99999,1.38422e-7,1.28718e-9]
(0.3,0.1)	[7.48245e-10,7.48246e-10,0.99999,1.38422e-7,1.28718e-9]
(0.2,0.155)	[5.2239e-10,5.22348e-10,0.9999,4.00815e-6,1.7466e-9]

Table 4: $\mathcal{U}^{M}$ Set optimized portfolios

We plot figures of the estimates in the Appendix (for $\alpha = 0.2, 0.2, \epsilon = 0.3, 0.2$). Looking at the weight percentages that are assigned to each of the 5 different assets, we see that the $\mathcal{U}^{CS}$ set assigns weights more uniformly throughout the portfolio, while the $\mathcal{U}^{M}$ set tends to place most of the weight on one asset. We can explain the phenomenon shown since $\mathcal{U}^M$ does not use the joint distribution and it therefore does not see benefits to diversification[2]. As a consequence, it invests all its wealth in the asset that appears to have the best worst-case quantile given the data. In contrast, $\mathcal{U}^{CS}$ is able to learn the covariance structure and from this, it is able to diversify across the assets, leading to a more evenly diversified portfolio.

3.2 Analysis and choosing models/parameters via cross-validation

Cross-validation is used to evaluate the fit of the model. The procedure consists of the following steps:

1. Partition data into training, validation, and test sets
2. Find optimal model on the training set and use the test set to check its predictive capability
3. See how well the model works on the test set to give the validation set
4. The validation error gives unbiased estimates of the power of the model to predict

Time-series are sometimes problematic for cross-validation. For example, suppose a pattern emerges in year 3 and stays for years 4-6, although it wasn't present in years 1 and 2.

So we choose an approach that is more well-suited for time series, *forward chaining*, with procedure:

- fold 1 : training [1], test [2]
- fold 2 : training [1 2], test [3]
- fold 3 : training [1 2 3], test [4]
- fold 4 : training [1 2 3 4], test [5]
- fold 5 : training [1 2 3 4 5], test [6]

This more accurately models the situation visible at prediction time, where you can model on past data and predict on forward-looking data and it also gives you a sense of the dependence of your model on size of the data. In summary, the training set should not contain information that occurs after the test set.

3.3 Cross-Validation Results

We let $k = 9$ since we have data in the range of a total of 9 years, 2007-2015.

Now we first use the training sample on the first 5 years. Then, we continue adding one more year to the training samples.

3.4 Calculations

To find the errors, we will employ a mean vector to represent a set of samples. The goodness of the model is assessed in terms of the MSE (mean-square-error) evaluated on the testing set, as used in k-fold cross-validation.

From the k-fold cross-validation error formula adapted to our method of choosing test and training sets in a forward chaining manner, for each $i = 1, ..., 9$ we fit our prediction function (corresponding to the oracles) on all points but those in the test fold, called $\hat{r}^{(i)}$ and evaluate the error on the points in the test fold.

$$CV_i(\hat{r}^{(i)}) = \frac{1}{n_k} \sum_{i \in F_{\text{test}}} (y_i - \hat{r}^{(i)}(x_i))^2$$

where n_k is the number of points in the test fold. Then we can average these fold-based errors to yield a test error estimate for the entire model,

$$CVErr(\hat{r}) = \frac{1}{k} \sum_{i=1}^{k} CV_i(\hat{r}^{(i)})$$

Now to choose between models, compute the cross-validated errors for each and then choose the model with the minimum cross-validated error.

So we have fitted the oracles based on data on the first 5 years, 6, 7, and 8 years. Then we can use these to solve the problem for the data in the test fold. This will give the set of $\hat{r}^{(i)}(x_i)$ which we will use with the original data (averaged to a vector of returns so we can compare columns where $n_k = 9$) to evaluate the error. Now we will list the calculated errors for each model, based on $\alpha, \epsilon = 0.3$.

Now, the average of these errors gives us CVErr($\mathcal{U}^{CS}$) = 0.02473, CVErr($\mathcal{U}^{M}$) = 0.12356. We can clearly see that CVErr($\mathcal{U}^{CS}$) < CVErr($\mathcal{U}^{M}$) by an approximate factor of 10, and therefore the model to use would be $\mathcal{U}^{CS}$.

For choosing optimal parameters α, ϵ, we assume that according to the definition of confidence regions that smaller values of the parameters (α, ϵ) yield more accurate results.

size of training data (yr)	value of $CV_i(\hat{r}^{(i)})$,$\mathcal{U}^M$	value of $CV_i(\hat{r}^{(i)})$,$\mathcal{U}^{CS}$
5	0.122413	0.023805
6	0.122504	0.02413677
7	0.1240449	0.0250294
8	0.1252704	0.02594487

Table 5: $CV_i(\hat{r}^{(i)})$ values for $\mathcal{U}^{CS}$,$\mathcal{U}^M$ model

4 Conclusion

In this project, two distinct optimization problems have been considered. We have formulated the robust formulations of the optimization problems, allowing the worst-case scenarios to be taken into consideration, under the uncertainty sets we picked.

In the robust knapsack problem, the data-driven uncertainty set $\mathcal{U}^{\chi^2}$ did not yield more efficient results than the polyhedral uncertainty set; and their solutions were both the same. This reflects a situation where the data-driven sets do not make a big impact on the outcome.

In the portfolio optimization problem, the min-max portfolio problem is of great interest, as interval uncertainty sets often produce portfolios with very low return, motivating the use of data-driven sets; while ellipsoidal sets reduce the problem to a standard model with a larger risk-aversion parameter[3]. In our experiment, the polyhedral uncertainty set did not yield a very meaningful solution, giving a vector of all zeros but for a single asset. While comparing data-driven sets, we saw through cross-validation that $\mathcal{U}^{CS}$ is a better choice of uncertainty set than $\mathcal{U}^M$, and this is explained through the way these sets were constructed. Since $\mathcal{U}^M$ does not use the joint distribution, it does not see the benefit to diversification of assets, and the asset it chooses may vary greatly depending on the subset of data used since it is difficult to estimate the best worst-case quantile on less samples of data. $\mathcal{U}^M$ does allocate to the asset that is shown to hold most of the returns. The cross-validation error that was given as a result of $\mathcal{U}^M$ was around 10 times greater than the error of $\mathcal{U}^{CS}$, confirming in [2] that $\mathcal{U}^{CS}$ is a good choice for this problem, and this experiment has shown that the theory behind data-driven uncertainty sets works. In the discrete optimization experiment, it has been verified that the set matches the same standards as the polyhedral uncertainty set. In the portfolio optimization experiment, the data-driven sets yielded results clearly more accurate and insightful than the polyhedral uncertainty set.

References

[1] Andrew Ang. *Asset Management*. Oxford University Press, August 2014.

[2] Nathan Kallus Dimitris Bertsimas, Vishal Gupta. Data-driven robust optimization. November 2014.

[3] Thomas F. Coleman Lei Zhu and Yuying Li. Min-max robust and cvar robust mean-variance portfolios. *Journal of Risk*, March 2009.

[4] Richard O. Michaud and Robert O. Michaud. Forecast confidence level and portfolio forecast confidence level and portfolio optimization. New Frontier Advisors' Newsletter, July 2004.

[5] CRSP Stocks. Available: Center for research in security prices. graduate school of business. university of chicago. Retrieved from Wharton Research Data Service, May 2016.

5 Appendix

5.1 Robust Formulation of Min-max Mean-variance Portfolio problem

In the Times Series optimization problem, in Equation 4, we can formulate the min-max robust formulation as follows:

$$\min_{x} \max_{\mu \in S_\mu, Q \in S_Q} -\mu^T x + \lambda x^T Q x \tag{8}$$

$$\text{s.t. } x \in \Omega \tag{9}$$

for uncertainty sets S_μ, S_Q for μ, Q.

In [3], they formulate the mean-variance portfolio problem based on the VaR and CVaR as a risk measure for the estimation risk.

CVaR as a risk measure is based on VaR, as a notion of worst case risk measure.

In more detail, consider a random variable R which denotes a specific risk, typically corresponding to loss. Assume that R has a density function $p(r)$.

The probability of R not exceeding a threshold α is:

$$\Phi(\alpha) = \int_{r \leq \alpha} p(r) dr, \tag{10}$$

where we have the assumption that the probability distribution for R has no jumps, and therefore Φ is everywhere continuous with respect to α.

Now given a confidence level $\beta \in (0, 1)$ (ex, $\beta = 95\%$) the associated Value-at-Risk is:

$$VaR_\beta = \min\{\alpha : \Phi(\alpha) \geq \beta\} \tag{11}$$

with the corresponding CVaR given by:

$$CVaR_\beta = \mathbb{E}(R | R \geq VaR_\beta) = \frac{1}{1-\beta} \int_{r \geq VaR_\beta} r p(r) dr. \tag{12}$$

Therefore, $CVaR_\beta$ is the expected loss conditional on the loss being greater than or equal to VaR_β and is used in [3].

By replacing the mean loss $-\mu^T x$ by a VaR measure of mean loss $VaR^\mu_\beta(-\mu^T x)$, they formulate the robust MV efficient portfolio problem's worst case scenario as,

$$\min_x VaR^{\mu}_{\beta}(-\mu^T x) + \lambda x^T Q x \tag{13}$$

$$\text{s.t. } x \in \Omega \tag{14}$$

and apply the following theoretical motivation connecting the relationship between VaR and CVaR with uncertainty sets from [2].

For any uncertainty set $\mathcal{U}$ constructed through the scheme, it implies a probabilistic guarantee for $\mathbb{P}$ at level ϵ, and therefore it must satisfy ([2] Proposition 1)

$$\phi_{\mathcal{U}}(x) \geq VaR^{\mathbb{P}}_{\epsilon}(x) \; \forall x \in \mathbb{R}^d.$$

and it has been shown in [2] that the choice of sets $\mathcal{U}^{CS}$,$\mathcal{U}^{M}$ provide the upper bound to this formulation above, and therefore we can write

$$\min_x \; -\mu^T x + \lambda x^T Q x \;\; \forall \mu \in VaR^{\mu}_{\beta} \subseteq \mathcal{U} \tag{15}$$

$$\text{s.t. } x \in \Omega. \tag{16}$$

5.2 Plots

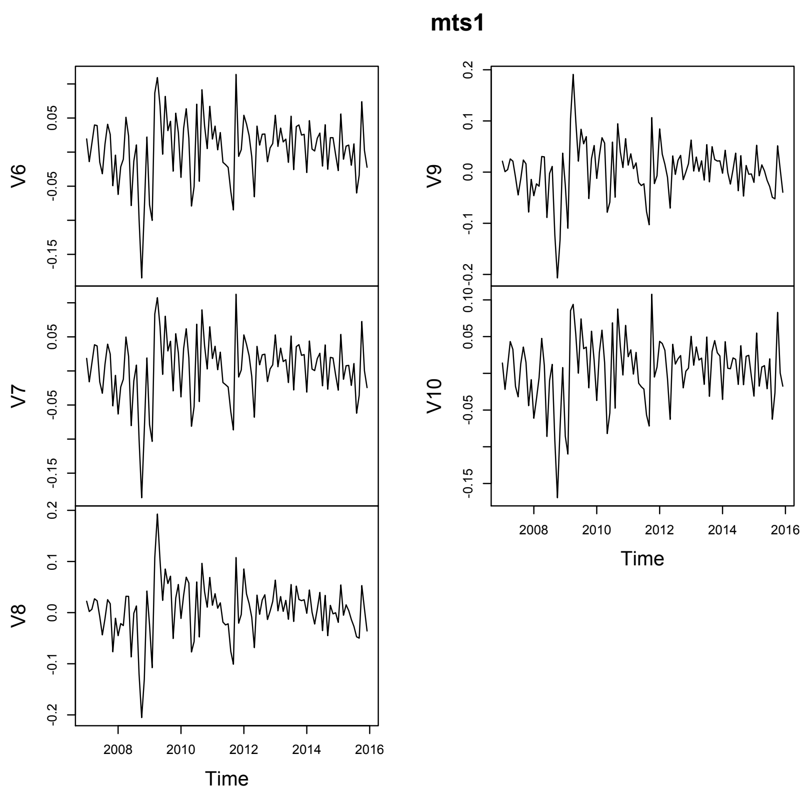

Figure 1: Time Series for Ebay Returns, 5 assets

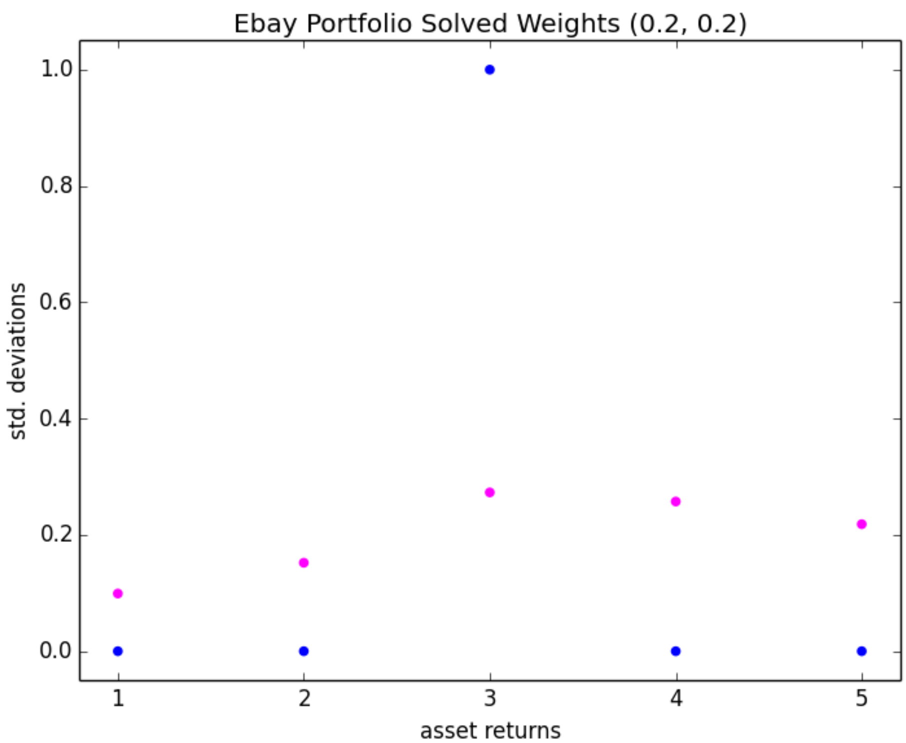

Figure 2: Optimal Weights of 5 assets as calculated by $\mathcal{U}^{CS}$ (magenta) and $\mathcal{U}^{M}$ (blue)

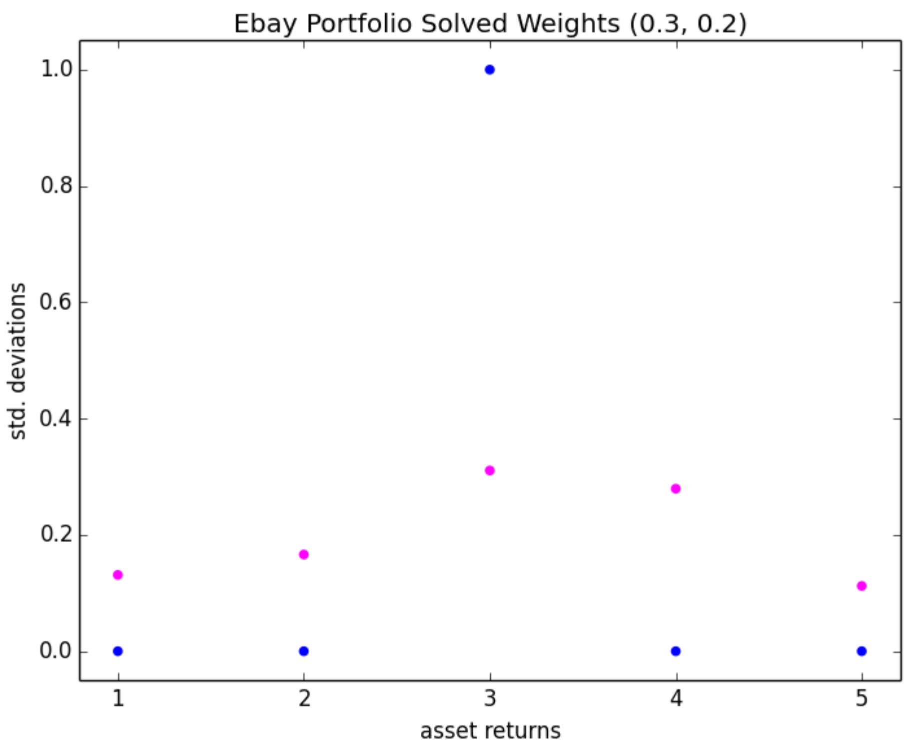

Figure 3: Optimal Weights of 5 assets as calculated by $\mathcal{U}^{CS}$ (magenta) and $\mathcal{U}^{M}$ (blue)

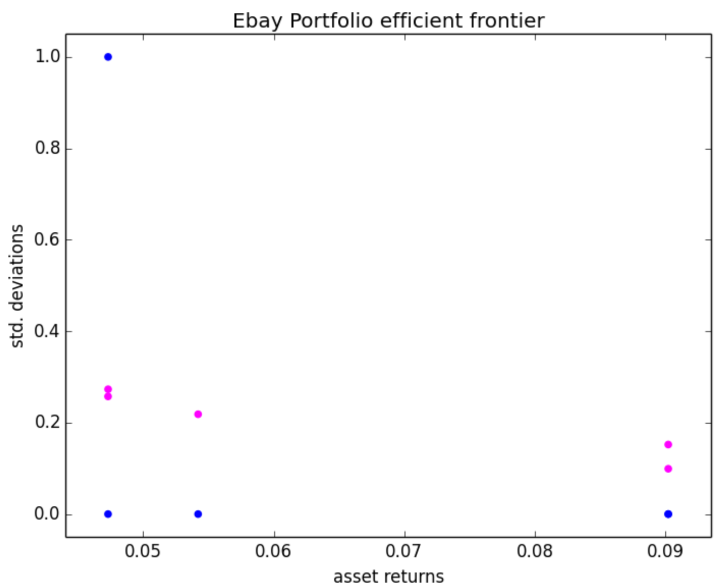

Figure 4: Ebay efficient frontier calculated by $\mathcal{U}^{CS}$ (magenta) and $\mathcal{U}^{M}$ (blue) for $(\alpha, \delta) = (0.2, 0.2)$

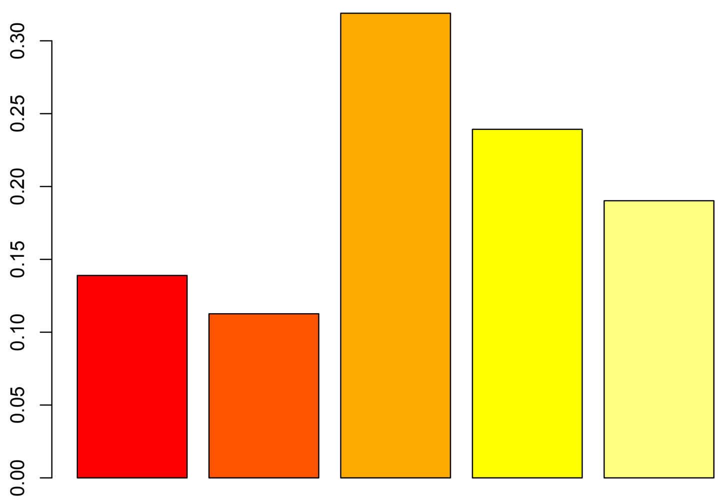

Figure 5: Actual Mean Returns Ebay Returns, 5 assets (averaged over years 2007-2015)