The Big Data Newsvendor Problem in a Portfolio Optimization Context

Thierry Bazier-Matte

Summer 2015

Abstract

Following [1], we provide a portfolio optimization method based on machine learning methods.

1 Introduction

This document considers a two-asset portfolio, of which one is the risk-free asset, yielding a constant return rate R_f, and the other being a risky asset s , typically a stock, yielding a random return rate r_{st} for each period t. We suppose that each risky asset s can be decribed daily by an *information vector* x_{st} containing potentially useful information, such as technical, fundamental or news-related information. Furthermore, we assume that the allocation of each asset of the portfolio p_{st} can be fully determined using a *decision vector* q. The allocation rule is the folowing: $q^T x_{st}$ is allocated to the risky asset and $1 - q^T x_{st}$ is allocated to the risk-free asset. Over the period t, the portfolio p_{st} consisting of asset s will therefore yield a return rate of:

Maybe considerations about the length of the period should be added? For example, it's not specified what's the period length of R_f.

$$p_{st}(q) = r_{st} q^T x_{st} + (1 - q^T x_{st}) R_f. \tag{1}$$

We further suppose that the *utility* derived from a return rate p can be described using a concave two-pieces linear function of the following form:

$$U(p) = p - r_c + \min((\beta - 1)(p - r_c), 0), \tag{2}$$

where r_c is an arbitrary *critical rate*. We impose the condition $0 < \beta < 1$, so that the utility derived from rates inferior to the critical rate r_c increases more sharply than from rates above r_c. Typically, the critical rate could be 0, but it could also be R_f. This is up to the investor.

More on it?

The question we now wish to ask is how the decision vector q should be chosen. We assume we have access to a training dataset S_n, comprising of s different assets over t periods, such that $n = s \times t$. Given such a training set, we then define the *optimal decision vector* $q^\star$ as the decision maximizing the average utility over the training set:

What's the difference between having n points with having $s \times t$ points? For example, what if $n \gg t$ or the reverse?

$$q^\star = \arg\max_q \hat{U} \tag{3}$$

$$= \arg\max_q \frac{1}{n} \sum_{s,t} U(p_{st}(q)). \tag{4}$$

The above optimization problem is linear and can therefore be readily solved with any modern computer.

However, we also wish to add a L_2 regularization term to the objective in order to avoid overfitting. The new learning algorithm is then

$$q^\star = \arg\max_q \frac{1}{n} \sum_{s,t} U(p_{st}(q)) - \lambda \|q\|^2. \tag{5}$$

Now even though this new problem is no longer linear, it is still convex and can therefore be efficiently solved.

We will refer to these two algorithms as respectively the non-regularized or linear algorithm and the regularized algorithm.

2 Bounds

Let $\boldsymbol{R}$ be the distribution of the returns of all considered assets. We suppose in this section that the distribution of absolute returns is bounded by $\bar{r}$.

Careful: distribution and space are not the same, but are treated equal.

Let also $\boldsymbol{X} \simeq \boldsymbol{R}^p$ be the *restricted information space* and $\boldsymbol{D} = \boldsymbol{R} \times \boldsymbol{X}$ be the *information space*. $\boldsymbol{D}^n$ is therefore the *observation space*, containing n observations.

The utility function being concave, we therefore have

$$\inf_{r \in \boldsymbol{R}} U(r) = -(\bar{r} + r_c) + \min((\beta - 1)(\bar{r} - r_c), 0) \tag{6}$$

We now try to determine the *algorithimically stable* bounds on the two algorithms derived in the previous section. We first start with the linear algorithm, namely, given a dataset S_n where, as before, $n = s \times t$, we want to find the largest (in the sup sense) theoretical difference between the utility observed when investing on a new datapoint $(x, r) \sim \boldsymbol{D}$ using a decision vector q_n formed from S_n and $q_{n'}$ formed from $S_n \setminus \{(x, r)\}$ where x is chosen at random. In other words, we simply try to observe how much the utility could be affected by a single observation. Obviously, the smaller the bound, the better it is since we avoid overfitting (the algorithm is less dependant on single points).

Mathematically, we want to bound the following expression, where $(x, r) \sim \boldsymbol{D}$. Here, we use the fact that the absolute first derivative of U is bounded by 1, and therefore is a 1-Lipschitz function[2]:

$$|U(q_n, x, r) - U(q_{n'}, x, r)| = |U(q_n^T x, r) - U(q_{n'}^T x, r)| \leq |(q_n - q_{n'})^T x|$$

Definition (Subgradient and subdifferential)**.** Form [3], a vector s is said to be a *subgradient* of a convex function f at a point x if, for all y,

$$f(y) \geq f(x) + s^T(y - x).$$

Geometrically, if s is a subgradient of f at x, then $h(z) = f(x) + s^T(z - x)$ is an hyperplane supporting the convex set epi f at $(x, f(x))$.

The set of all subgradients of f at x is the subdifferential and is denoted $\partial f(x)$.

TODO!

Reciprocally, we also define *supergradients* the same way as subgradients, but for concave functions[4], that is s is a supergradient to f at x_0 if and only if

$$f(x) \leq f(x_0) + s^T(x - x_0).$$

Because $\hat{U}$ is the average utility over all data points of S_n, $\hat{U}$ is a convex function of the decision vector q in $\boldsymbol{R}^p$, because U is itself convex in $\boldsymbol{R}^p$. Let $\partial\hat{U}$ be the set of supergradients of $\hat{U}$ at $q^\star$. Then, for any $s \in \partial\hat{U}$, the following hold by definition of the supergradient:

$$s^T(q_{n'} - q_n) \geq \hat{U}(q_{n'}, S_n) - \hat{U}(q_n, S_n). \tag{7}$$

In particular, it easy to see that $0 \in \partial\hat{U}$ because q_n is the optimal decision vector for the information dataset S_n. It follows that the following holds:

$$0 \geq s^{\star T}(q_{n'} - q_n) \geq \hat{U}(q_{n'}, S_n) - \hat{U}(q_n, S_n). \tag{8}$$

where

$$s^\star = \underset{s \in \partial\hat{U}}{\arg\min}\, s^T(q_{n'} - q_n) \tag{9}$$

References

[1] Cynthia Rudin and Gah-Yi Vahn. *The Big Data Newsvendor: Pratical Insights from Machine Learning*, Operations Research, 2015.

[2] "Si la valeur absolue de la dérivée est majorée par k, f est k-lipschitzienne". Application lipschitzienne.

[3] Rockafellar, R. T. *Convex Analysis*, Princeton University Press, 1970.

[4] Supergradients.

[5] Reference needed!