sparse index tracking proposal

Shane Chu

September 2021

1 Abstract

Sparse Index Tracking is a well-known passive portfolio management strategy in finance, that aims to capture the value from an index (e.g. SP500) using as few assets as possible. This problem can be formulated as a sparse signal discovery problem, where the sparsity inducing function is typically set to be the ℓ_0 norm. The most popular method for solving the canonical form of such problem is through a majorization-minimization (MM) approach, as shown in [BP18]. In this project, we seek to use other formulations of the sparse index tracking problem and carry out different algorithmic solutions to examine the time/space complexity trade-offs and the prediction results.

2 Why sparsity?

We can track the SP500 index trivially by buying all the assets as listed in the SP500 index. However, doing so is impractical for an asset manager, as it might be costly due to the commission fees, and other transaction costs, etc. In more general settings, some assets that are listed under an index may be illiquid, and it may be the portfolio manager's best interest to avoid such assets as much as possible.

3 Problem formulation

3.1 Notations

We assume that an index is composed of N assets. Moreover,

- The net return of the index is a vector $\mathbf{r^b} = \left[r_1^b, \ldots, r_T^b\right]^T \in \mathbb{R}^T$
- The data matrix $\mathbf{X} = [\mathbf{r}_1, \ldots, \mathbf{r}_T]^T \in \mathbb{R}^{T \times N}$, which shows the net return of N assets in the past T days
  - The vector $\mathbf{r_t} \in \mathbb{R}^N$ for $t = 1, \ldots, T$ denotes the net returns of the N assets at the t-th day.
- The vector $\mathbf{b_t} \in \mathbb{R}^N_{++}$ for $t = 1, \ldots, T$, is the normalized benchmark index weights at the t-th day, such that $\mathbf{b_t}^T\mathbf{1} = 1$ and $\mathbf{r_t}^T\mathbf{b_t} = r_t^b$.
- The designed portfolio vector $\mathbf{w} \in \mathbb{R}^N_+$ such that $\mathbf{w}^T\mathbf{1} = 1$.

3.2 The problem

The goal is that we want to find $\mathbf{w}$ such that $\mathbf{Xw} \approx \mathbf{r^b}$, where $\mathbf{w}$ is a sparse vector. Two error functions that are commonly employed are the *empirical-tracking-error*, which is defined as

$$ETE(\mathbf{w}) = \frac{1}{T}\left\|\mathbf{r^b} - \mathbf{Xw}\right\|_2^2$$

and the *downside-risk*, which is defined as

$$DR(\mathbf{w}) = \frac{1}{T}\| \left(\mathbf{r^b} - \mathbf{Xw}\right)_+ \|_2^2$$

where $(\cdot)_+$ zeros out the negative entries in the vector, as in downside-risk we are only interested when the index beats our portfolio. The sparse index tracking problem is formulated as

$$\begin{aligned} \underset{\mathbf{w}}{\operatorname{argmin}} \quad & TE(\mathbf{w}) + \lambda \|\mathbf{w}\|_0 \\ \text{s.t.} \quad & \mathbf{w}^T \mathbf{1} = 1 \\ & \mathbf{w} \geq 0 \end{aligned} \tag{1}$$

where TE is set to be either $ETE(\cdot)$ or $DR(\cdot)$. Note that the typical approach such as the one in [BP18], the ℓ_0 norm is replaced by a continuous and differentiable function $\rho_{\mu,p}(w) = \frac{\log(1+w/p)}{\log(1+\mu/p)}$, so now (1) becomes

$$\begin{aligned} \underset{\mathbf{w}}{\operatorname{argmin}} \quad & TE(\mathbf{w}) + \lambda \mathbf{1}^T \rho_{\mu,p}(\mathbf{w}) \\ \text{s.t.} \quad & \mathbf{w}^T \mathbf{1} = 1 \\ & \mathbf{w} \geq 0 \end{aligned} \tag{2}$$

where $\rho_{\mu,p}(\mathbf{w}) = (\rho_{\mu,p}(w_1), \dots, \rho_{\mu,p}(w_N))$.

3.3 Issues with ℓ_1 norm

In (1), as we set the constraint to be $\mathbf{w}^T\mathbf{1} = 1$, the ℓ_1-norm becomes irrelevant because $\|\mathbf{w}\|_1 = \sum_{i=1}^N |w_i| = \sum_{i=1}^N w_i = 1$.

4 Possible directions to explore

4.1 ADMM

A relatively straight-forward way to solve (1) is to use alternating direction method of multipliers (ADMM) [BPC11], where [Bec17] has provided many useful shortcuts for the derivations for the proximal updates. As ADMM is suitable for large-scale optimization tasks, we can compare it to the traditional MM approach to see which method is more suitable for large data set.

4.2 Structured sparsity

As shown in [RG14], sometimes we want more "structure" in the sparse vector induced by the sparse regularization, as sparse inducing function such as ℓ_1-norm might tend to induce the sparsity homogeneously across the entire vector. This may be resolved by replacing the sparsity inducing function by "group norms" as shown in [RG14], provided that we have some more prior information on the data.

4.3 Escaping the saddle points

The problem (1) is a non-convex problem due to the choice of ℓ_0 regularization. Techniques such as adding noise in the gradient as shown in [Jin+17] could be carried out and see whether there are noticeable difference in the solutions.

5 Problem

Last time we saw that the sparse index tracking problem can be formulated as

$$\begin{aligned} \underset{w}{\operatorname{argmin}} \quad & \frac{1}{T} \left\| r^b - Xw \right\|_2^2 + \lambda \|w\|_0 \\ \text{s.t.} \quad & w^T \mathbf{1} = 1 \\ & w \geq 0 \end{aligned} \tag{3}$$

Let's write it in an unconstrained fashion. A way to do so to (1) is:

$$\underset{w}{\operatorname{argmin}} \quad \frac{1}{T}\left\|r^b - Xw\right\|_2^2 + \lambda\|w\|_0 + \mathbb{1}_C(w) \tag{4}$$

where $\mathbb{1}$ is the indicator function (i.e. $\mathbb{1}(x) = 0$ if $x \in C$ else $\mathbb{1}(x) = \infty$) and C is the constraint set, i.e. $C = \{w : w^T\mathbf{1} = 1, w \geq 0\}$[1]. We can solve the problem using alternating direction method of multipliers (ADMM). To do so, we first apply a variable splitting to introduce auxiliary variables v_1, v_2, v_3 to restate (2) as follows:

$$\begin{aligned} \underset{w}{\operatorname{argmin}} \quad & \frac{1}{T}\left\|r^b - v_1\right\|_2^2 + \lambda\|v_2\|_0 + \mathbb{1}_C(v_3) \\ \text{s.t.} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} &= \begin{bmatrix} X \\ I \\ I \end{bmatrix} w \end{aligned} \tag{5}$$

Let us write the constraint as $Aw + Bv = 0$, as we observe that we can set $A = \begin{bmatrix} X^T & I & I \end{bmatrix}^T$, $B = -I$, and $v = \begin{bmatrix} v_1^T & v_2^T & v_3^T \end{bmatrix}^T$. Now, the augmented Lagrangian in scaled form (See 3.1.1 in [BPC11]), as commonly used in the ADMM framework, is:

$$L_\rho(w, \eta) = \frac{1}{T}\left\|r^b - v_1\right\|_2^2 + \lambda\left\|v_2\right\|_0 + \mathbb{1}_C(v_3) + \frac{\rho}{2}\left\|Aw + Bv + u\right\|_2^2 \tag{6}$$

where u is the (scaled) Lagrange multipliers, i.e. $u = \begin{bmatrix} u_1^T & u_2^T & u_3^T \end{bmatrix}^T$, and ρ is the penalty parameter.

6 ADMM

With (4) at hand, the ADMM is

$$w^{(i+1)} := \underset{w}{\operatorname{argmin}} \frac{\rho}{2}\left\|Aw + Bv^{(i)} + u^{(i)}\right\|_2^2 \tag{7}$$

$$v_1^{(i+1)} := \underset{v_1}{\operatorname{argmin}} \left\{\frac{1}{2}\|r^b - v_1\|_2^2 + \frac{\rho}{2}\left\|v_1 - \left(Xw^{(i+1)} + u_1^{(i)}\right)\right\|_2^2\right\} \tag{8}$$

$$v_2^{(i+1)} := \underset{v_2}{\operatorname{argmin}} \left\{\lambda\|v_2\|_0 + \frac{\rho}{2}\left\|v_2 - \left(w^{(i+1)} + u_2^{(i)}\right)\right\|_2^2\right\} \tag{9}$$

$$v_3^{(i+1)} := \underset{v_3}{\operatorname{argmin}} \left\{\mathbb{1}_C(v_3) + \frac{\rho}{2}\left\|v_3 - \left(w^{(i+1)} + u_3^{(i)}\right)\right\|_2^2\right\} \tag{10}$$

$$u^{(i+1)} := u^{(i)} + Aw^{(i+1)} + Bv^{(i+1)} \tag{11}$$

where superscripts i's are the iteration indices. A straight-forward calculation shows that (5) is solved by taking the solution of the following linear system:

$$\left(X^TX + 2I\right)w = X^T\left(v_1^{(i)} - u_1^{(i)}\right) + \left(v_2^{(i)} - u_2^{(i)}\right) + \left(v_3^{(i)} - u_3^{(i)}\right) \tag{12}$$

Similarly, the solution to (6) is:

$$\frac{1}{\rho + 1}\left(r^b + \rho\left(Xw^{(i+1)} + u_1^{(i)}\right)\right) \tag{13}$$

The solution to (7) is:

$$\mathbf{H}_{\sqrt{2\frac{\lambda}{\rho}}}\left(w^{(i+1)} + u_2^{(i)}\right) \tag{14}$$

[1]Abuse the inequality $\geq$ notation to write $w \geq 0$ to mean each component of w is nonnegative.

where $\mathbf{H}_{\sqrt{2\frac{\lambda}{\rho}}}$ the hard-threshold operator (example 6.10, [Bec17]). Hard-threshold operator is applied element-wise to the K-dimensional input vector x as $\mathbf{H}_\alpha(x) = (H_\alpha(x_1), \cdots, H_\alpha(x_K))$, where $H_\alpha(v_k) = v_k$ if $|v_k| > \alpha$ else 0 (in the case $|v_k| = \alpha$, both $v_k, 0$ can be used as solution). Last, to solve (8), note that the constraint set C is an intersection of a hyperplane and and the non-negative orthant. Let $P_{w\geq 0}$ be the projection operator that projects w to the non-negative orthant. The solution to (8) is then

$$P_{w\geq 0}\left(\left(w^{(i+1)} + u_3^{(i)}\right) - \mu^*\mathbf{1}\right) \tag{15}$$

where μ^* is a scalar that is the solution of the equation

$$\mathbf{1}^T P_{w\geq 0}\left(\left(w^{(i+1)} + u_3^{(i)}\right) - \mu\mathbf{1}\right) = 1 \tag{16}$$

As noted in remark 6.28 in [Bec17], (14) can be solved with simple bisection procedure.

7 Notes

The computationally expensive step is (10), which requires cubic time if solved by Gaussian elimination. We may need to investigate whether special structure can be found in X so that further speed up is possible.

References

[BPC11] Stephen Boyd, Neal Parikh, and Eric Chu. *Distributed optimization and statistical learning via the alternating direction method of multipliers*. Now Publishers Inc, 2011.

[RG14] Irina Rish and Genady Grabarnik. *Sparse modeling: theory, algorithms, and applications*. CRC press, 2014.

[Bec17] Amir Beck. *First-order methods in optimization*. SIAM, 2017.

[Jin+17] Chi Jin et al. "How to escape saddle points efficiently". In: *International Conference on Machine Learning*. PMLR. 2017, pp. 1724–1732.

[BP18] Konstantinos Benidis and Daniel P Palomar. "Design of Portfolio of Stocks to Track an Index". In: (2018).