Active vs passive portfolio management

Active: Beat a benchmark

- Maximize alpha
- Remain "not-too-far" from the benchmark
- Various types of constraints: size of active holdings, tracking error, attributes such as beta, sector exposures, etc.

Passive: Track a benchmark as simply as possible

- Minimize tracking error
- Match or approximate various benchmark attributes such as beta, sector exposures, etc.
- Use a parsimonious portfolio: include only a few stocks
- We will next discuss a possible approach to this problem

Cardinality and threshold constraints

- Limit on the number of stocks in a portfolio
- Lower bound on portfolio's positions
- These can be modeled via *binary variables*
- The optimization problem is no longer convex and becomes substantially harder to solve
- Numerical solvers use heuristics and enumeration ("branch and bound", "branch-and-cut")

Construction of an index fund

Problem

Track a benchmark with few (say q) stocks.

A possible formulation

$$\begin{array}{ll} \min & (\mathbf{x}-\mathbf{x}_B)^\mathsf{T} V(\mathbf{x}-\mathbf{x}_B) \\ \text{s.t.} & \mathbf{1}^\mathsf{T}\mathbf{x} = 1 \\ & \mathbf{x} \geq 0 \\ & \text{"at most } q \text{ of the } x_j \text{ are positive"} \end{array}$$

How can we formulate this *cardinality* constraint?

Bad news:

- These kinds of optimization problems (with integrality constraints) are very difficult to solve.
- Basic approach (branch and bound):
  – Ignore integer variables and see if that works.
  – If not, generate two new problems and keep trying.
- This procedure can be enhanced, but for some problems it may need an astronomically large number of steps before it finds the optimal solution.

Alternative:

Use heuristics that generate "reasonably good" solutions.

Back to the index fund problem

Alternative 1

Use linear programming instead of quadratic programming.

Alternative 2

- First, pick a set of q stocks (via some kind of clustering)
- Build a portfolio with the chosen stocks

A neat fact about linear programming

Consider a simple linear program

$$\begin{array}{ll} \max & \mathbf{c}^{\mathsf{T}}\mathbf{x} \\ \text{s.t.} & A\mathbf{x} = \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0} \end{array}$$

where $A \in \mathbb{R}^{m \times n}, \mathbf{c} \in \mathbb{R}^n, \mathbf{b} \in \mathbb{R}^m$ with $m \leq n$.

If this linear program has an optimal solution, then it has one with at most m positive components.

Possible approach to index-fund problem:

Construct a portfolio that matches the benchmark on several (around q) attributes.

Index fund problem again

Alternative 2

- First, pick a set of q stocks (via some kind of clustering)
- Build a portfolio with the chosen stocks:
  - Set portfolio weights according to the benchmark size of the clusters
  - Set up a new optimization problem to best track the index

Stock clustering problem

- Set of n stocks
- For stocks i, j have a similarity measure ρ_{ij}
- Want to choose a set of q stocks so that each one of the n stocks is "similar" to one of the q chosen stocks.

How can we formulate this problem?

Integer programming formulation to stock clustering

Assume

ρ_{ij} : Similarity between stock i and j

Formulation (cont)

$$
\begin{array}{lll}
\max\limits_{\mathbf{x},\mathbf{y}} & \displaystyle\sum_{i=1}^{n}\sum_{j=1}^{n} \rho_{ij} x_{ij} & \\
& \displaystyle\sum_{j=1}^{n} y_j = q & \\
& \displaystyle\sum_{j=1}^{n} x_{ij} = 1 & \text{for } i = 1, \ldots, n \\
& x_{ij} \leq y_j & \text{for } i, j = 1, \ldots, n \\
& x_{ij}, y_j & \text{binary for } i, j = 1, \ldots, n.
\end{array}
$$

Although this is a difficult problem, there is a clever heuristic that gives pretty good (often optimal or nearly optimal) solutions.

Heuristic for stock clustering problem

Lagrangian relaxation by Cornuéjols et al (see notes).

Files "fund.cpp" and "fund.exe":

- C++ implementation of the above Lagrangian-based algorithm (by F. Margot).
- Need to input data in a suitable form

$$\begin{array}{cccc} n & q & & \\ 1 & 1 & \cdots & 1 \\ \rho_{11} & \rho_{12} & \cdots & \rho_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ \rho_{n1} & \rho_{n2} & \cdots & \rho_{nn} \end{array}$$

  The ρ_{ij} need to be scaled up and rounded so that they are integer.
- Get a heuristic solution $\mathbf{y}$ and $\mathbf{x}$ together with an upper bound for the stock clustering problem in an output file.