IEOR 4739: A first-order algorithm for mean-variance QPs

The problem we are interested in is as follows:

$$\text{Minimize } F(x) \ \doteq\ \lambda \left(\sum_i \sigma_{ii}^2 x_i^2 \ +\ 2 \sum_{i<j} \sigma_{ij} x_i x_j \right) \ -\ \sum_i \mu_i x_i,$$

Subject to the constraints:

$$\sum_j x_j \ =\ 1, \quad \text{and}$$

$$l_j \ \leq\ x_j \ \leq\ u_j, \quad \text{for all } j.$$

Step 1. Achieve feasibility

This works as follows: we *start* with $\bar{x}_i = l_i$ for all i. If it is already the case that $\sum_i \bar{x}_i > 1$ then we can quit because the problem is infeasible (no solutions exist). If $\sum_i \bar{x}_i = 1$ (or very close to it) then our $\bar{x}$ is feasible. In the remaining case, $\sum_i \bar{x}_i < 1$, we simply increase each $\bar{x}_i$, *one at a time*, starting with $\bar{x}_1$, then $\bar{x}_2$, and so on, until we achieve $\sum_i \bar{x}_i = 1$. We stop the increase of any $\bar{x}_i$ when it reaches its upper bound, u_i. If it is the case that each $\bar{x}_i = u_i$ and we *still* have $\sum_i \bar{x}_i < 1$ then again we quit since the problem is infeasible.

Step 2. Improvement phase

In the improvement phase, we will repeatedly try to improve the solution. For $k = 1, 2, \ldots$ we will compute, from $x^{(k)}$, a new feasible vector $x^{(k+1)}$ such that

$$F(x^{(k+1)}) \ <\ F(x^{(k)}).$$

This computation will be done in two parts.

(k.1) Compute a vector $y^{(k)}$ such that

$$x^{(k)} + y^{(k)} \text{ is feasible, i.e.} \quad \sum_j (x_j^{(k)} + y_j^{(k)}) = 1 \text{ and } l_j \ \leq\ x_j^{(k)} + y_j^{(k)} \ \leq\ u_j \text{ for all } j. \tag{1}$$

and

$$\left[\nabla F(x^{(k)}) \right]^T y^{(k)} \ <\ 0, \tag{2}$$

if such a vector $y^{(k)}$ exists. (Q: what if it does not exist?)

(k.2) Compute a value $0 < s^* < 1$ such that

$$s^* \quad = \quad \text{argmin} \left\{ F(x^{(k)} + s y^{(k)}) \ :\ 0 \leq s \leq 1 \right\}. \tag{3}$$

Then we set $x^{(k+1)} \ =\ x^{(k)} + s y^{(k)}$.

Let's review what the algorithm does. Condition (2) guarantees that if we move a small distance along the vector $y^{(k)}$ (starting from $x^{(k)}$ the function $F(x)$ will decrease. The question is *how far* we should move. Condition (1) guarantees that any step size between 0 and 1 will yield a feasible point, so we only need to make sure that we choose the stepsize so as to minimize $F(x)$ along the step direction. This is what (3) does.

To completely specify the algorithm we need to explain how the search in step (k.2) is performed, and also how to compute $y^{(k)}$ in (k.1). Let's first consider the step size search. This is easy because

$$G(s) \ \doteq\ F(x^{(k)} + s y^{(k)})$$

is a degree-2 polynomial in s, so we simply need to find s with $G'(s) = 0$ (and check the boundary conditions for s). "Details left to the reader."

Now we turn to step (k.1). Write

$$g_k \doteq \nabla F(x^{(k)}).$$

Then to carry out (k.1) we just need to solve the linear program

$$\min \quad g_k^T y \tag{4a}$$

Subject to

$$\sum_j y_j = 0 \tag{4b}$$

$$l_j - x_j^{(k)} \leq y_j \leq u_j - x_j^{(k)} \quad \text{all } j. \tag{4c}$$

Now this is a very simple linear program and there is a simple algorithm for it. In fact, assume that the asset indices have been sorted so that

$$g_{k,1} \geq g_{k,2} \geq \cdots g_{k,n}.$$

(Note: in general this sorting will depend on the iteration index k). **Then:** there is an index m such that the optimal solution y^* to (4) is given by

$$y_j^* = \begin{cases} l_j - x_j^{(k)} & j < m \\ u_j - x_j^{(k)} & j > m \end{cases} \tag{5}$$

(Convince yourself that this is true – it does take some effort!) Note that this formula does not specify y_m^*. However since $\sum_j y_j^* = 1$ this quantity is easily determined.

In summary, then, we solve the auxiliary linear program by *enumerating* all possible values for m. Each such value will give us one candidate vector y^* – which may be infeasible, in which case it is rejected. We simply keep the best feasible candidate (Q. how many feasible candidates will there be?).

Fri.Jan.29.210138.2016@rockadoodle