1 Subroutine for SDForest

We are fitting a single tree. At step m, the tree has m leaves. We encode this by a matrix $E \in \mathbb{R}^{n\times m}$. We write $e_1, \ldots, e_m$ for the columns of E. The matrix E has the property that if $1 \le l < t \le m$, either e_l and e_t have disjoint support or the support of e_t is contained in the support of e_l. To find the best $(m+1)$th split, we consider a large number of candidate split encoded by a new column e_{m+1}, which has a 1 in the ith entry if the ith sample point x_i lies in the new leaf. We want to find the new column such that $\|QE_{m+1}\hat{\beta}_{m+1} - QY\|_2^2$ is minimal among the candidate splits, with $E_{m+1} = (E_m, e_{m+1}) \in \mathbb{R}^{n\times(m+1)}$ and $\hat{\beta}_{m+1}$ is the least squares estimator of QY vs. QE_{m+1}. The goal of this note is to show, how we can efficiently find the best split e_{m+1} without having to estimate a linear model "from scratch" each time.

By induction, assume that we are given a QR-decomposition of the matrix QE_m, i.e. there exists a matrices $U_m \in \mathbb{R}^{n\times m}$ and $R_m \in \mathbb{R}^{m\times m}$ such that the columns $u_1, \ldots, u_m$ of U_m are orthonormal and R_m is an upper triangular matrix and

$$QE_m = U_m R_m.$$

For a candidate split encoded by e_{m+1}, let $w_{m+1} = Qe_{m+1}$. Define

$$u'_{m+1} = w_{m+1} - (w_{m+1}^T u_1)u_1 - \ldots - (w_{m+1}^T u_m)u_m.$$

Then, define $u_{m+1} = u'_{m+1}/\|u'_{m+1}\|$. Note that Note that u_{m+1} is orthogonal to $u_1, \ldots, u_m$. Moreover, w_{m+1} is in the span of $u_1, \ldots, u_{m+1}$ and $w_{m+1} = (w_{m+1}^T u_1)u_1 + \ldots + (w_{m+1}^T u_{m+1})u_{m+1}$. Define $r_{m+1} = (w_{m+1}^T u_1, \ldots, w_{m+1}^T u_{m+1})^T \in \mathbb{R}^{m+1}$. In total, we can write

$$QE_{m+1} = U_{m+1} R_{m+1},$$

where U_{m+1} has orthonormal columns $u_1, \ldots, u_{m+1}$ and

$$R_{m+1} = \begin{pmatrix} R_m & r_{m+1} \\ 0 & \vdots \end{pmatrix}$$

is an upper triangular matrix. The least squares estimator $\hat{\beta}_{m+1} = \arg\min_\beta \|QE_{m+1}\beta - QY\|^2$ is given by

$$\begin{aligned}
\hat{\beta}_{m+1} &= ((QE_{m+1})^T QE_{m+1})^{-1}(QE_{m+1})^T QY \\
&= (R_{m+1}^T U_{m+1}^T U_{m+1} R_{m+1})^{-1} R_{m+1}^T U_{m+1}^T QY \\
&= R_{m+1}^{-1} U_{m+1}^T QY
\end{aligned}$$

We are interested in choosing e_{m+1} such that $\|QE_{m+1}\hat{\beta}_{m+1} - QY\|_2^2$ is minimal. Note that

$$\begin{aligned}
\|QE_{m+1}\hat{\beta}_{m+1} - QY\|_2^2 &= \|U_{m+1}R_{m+1}R_{m+1}^{-1}U_{m+1}^TQY - QY\|_2^2 \\
&= \|U_{m+1}U_{m+1}^TQY - QY\|_2^2 \\
&= (QY)^T(I - U_{m+1}U_{m+1}^T)^2QY \\
&= (QY)^T(I - U_{m+1}U_{m+1}^T)QY \\
&= \|QY\|^2 - \|U_{m+1}^TQY\|^2 \\
&= \|QY\|^2 - \|U_m^TQY\|^2 - (u_{m+1}^TQY)^2
\end{aligned}$$

Hence, we need to choose e_{m+1} such that $(u_{m+1}^TQY)^2$ is maximal.

Hence, the algorithm to find the optimal split e_{m+1} has the following steps:

For all candidate splits s, let e_{m+1}^s be the encoding of this split. For all s, do

1. $w_{m+1} = Qe_{m+1}^s$
2. $u'_{m+1} = w_{m+1} - (w_{m+1}^Tu_1)u_1 - \ldots - (w_{m+1}^Tu_m)u_m$
3. Store $\alpha_s = (u'^T_{m+1}QY)^2/\|u'_{m+1}\|^2$

Choose s, such that α_s is maximal. Then save $u_{m+1} = u'_{m+1}/\|u'_{m+1}\|$ with the u'_{m+1} from the optimal s.

This can again be made faster (note $Q^T = Q$): For all candidate splits s, let $e_{m+1} = e_{m+1}^s$ be the encoding of this split. For all s, do

1. $u'_{m+1} = Qe_{m+1} - (e_{m+1}^TQu_1)u_1 - \ldots - (e_{m+1}^TQu_m)u_m$
2. Store $\alpha_s = (u'^T_{m+1}QY)^2/\|u'_{m+1}\|^2$

Choose s, such that α_s is maximal. Then save $u_{m+1} = u'_{m+1}/\|u'_{m+1}\|$ with the u'_{m+1} from the optimal s. This should be faster since $Qu_1, \ldots, Qu_m$ only have to be calculated once.

This can again be rewritten. We use $(e_{m+1}^TQu_j)u_j = u_ju_j^TQe_{m+1}$. Hence, we can replace the first line by

1. $u'_{m+1} = \left(Q - \sum_{l=1}^m u_lu_l^TQ\right)e_{m+1}$

Hence, $\left(Q - \sum_{l=1}^m u_lu_l^TQ\right)$ is always the same and only needs to be updated by subtracting $u_{m+1}u_{m+1}^TQ$, once the best split is decided. Hence, what is remained to do for each candidate split is really just the matrix vector product $\left(Q - \sum_{l=1}^m u_lu_l^TQ\right)e_{m+1}$ and the scalar product $(u'^T_{m+1}QY)^2/\|u'_{m+1}\|^2$ in step 2. This should be faster than the approach with solving a linear model if the second step can be made sufficiently fast.