This draft describes a procedure for an efficient optimization of ridge penalties for an optimal OOS CV performance. We define the loss function for the problem, get analytical expressions for the gradient and the Hessian, and specify how the computation needs to go, including the quantities that should be precomputed.

Suppose that we're using ridge to fit endogenous $Y \in \mathbb{R}^{n \times n_y}$ to exogenous $X \in \mathbb{R}^{n \times n_x}$. In the standard ridge setting with penalty a the in-sample fit is $\hat{Y} = X(X^T X + aI)^{-1} X^T Y$, with I representing an identity matrix. For simplicity's sake, we assume that Y and X are centered, and we can thus ignore the intercept (i.e., the intercept effectively gets no penalty). If we also introduce sample weighting $w \in \mathbb{R}^n$, the solution becomes:

$$\hat{Y}(X, Y, w, a) = Xb,\ b := h\, v,\ h := g^{-1},\ g := u + aI,\ u := X^T \mathrm{diag}(w) X,\ v := X^T \mathrm{diag}(w) Y,$$

To choose a in practice for a particular univariate target $y \in \mathbb{R}^n$ one may do a train-test split and solve the following optimization problem:

$$a := \mathrm{argmin}_a \sum_{I,J \in \text{train-test splits}} \tilde{w}_J (y_J - \tilde{X}_J b(X_I, y_I, \bar{w}_I, a))^2, \tag{1}$$

where the test data is disjoint, $\tilde{w}_J$ is normalized to sum up to 1 over the test data, and each individual $\bar{w}_I$ sums to 1.

Suppose that we allow varying penalties $a \in \mathbb{R}^{n_x}$, and we want to optimize them. We can write the loss, gradient and Hessian for each summand in the problem (1) explicitly. We define $\partial_{i,j} f$ to be the i^{th} row and j^{th} column of the Hessian of f w.r.t. a_i and a_j, and the bar refers to the test data. The loss that we optimize: $\sum_{i,j} \bar{w}_i (\bar{y}_i - \bar{x}_{i,j} b_j)^2$. It's gradient ∂_k is:

$$-2 \sum_i \bar{w}_i \sum_j \bar{x}_{i,j} (\bar{y}_i - \bar{x}_{i,j} b_j) \partial_k b_j.$$

Since $gb = v$, $\partial_k v_i = 0$ and $\partial_k g_{i,j} = \delta_{i,j,k}$, we have

$$b_k + \sum_j g_{i,j} \partial_k b_j = 0 \Rightarrow \partial_k b_j = -h_{j,k} b_j.$$