1 Implementation of nonparametric logistic regression

We have implemented non-parametric logistic regression in C++ following the scheme outlined in Green and Silverman [1994].

Our target and decoy PSMs are divided into N bins. For each bin i, we record the total number of scores m_i, the median score $\hat{x_i}$, and the total number of decoy scores y_i in the bin. We model our observations of target and decoy PSMs as outcomes from binomial processes with probability $p_i = p(\hat{x_i})$ for a decoy PSM, so that $Y_i \sim B(m_i, p_i)$. We use a non-linear link function $g(x)$, which we model with a natural cubic spline $\hat{g}(x)$. We assume that the spline takes the values g and the splines second derivative takes the values γ at the spline knots $\hat{x}$.

Before describing the procedure, let us first define two matrices, Q and R, that will remain invariant through the procedure. The elements of the $n \times (n-2)$ matrix Q are defined as

$$q_{i,j} = \begin{cases} h_{j+1}^{-1} & : \quad i = j \\ -h_{j+1}^{-1} - h_{j+2}^{-1} & : \quad i = j+1 \\ h_{j+2}^{-1} & : \quad i = j+2 \\ 0 & : \quad \text{otherwise} \end{cases} \tag{1}$$

where $h_i = \hat{x}_{i+1} - \hat{x}_i$. The elements of the $(n-2) \times (n-2)$ matrix R are defined as

$$r_{i,j} = \begin{cases} \frac{1}{3}h_{i-1} + \delta x_i) & : i = j \\ \frac{1}{6}h_i & : j = i+1 \\ \frac{1}{6}h_j & : j+1 = i \\ 0 & : \text{otherwise} \end{cases} . \tag{2}$$

The matrices Q and R define a relationship between any natural cubic spline having spline knots at $\hat{x}$ and its second derivative:

$$Q^T g = R\gamma.$$

Let us also define two vectors and a matrix that, unlike the matrices above, get updated during for each iteration. The vector z is defined as

$$z_i = g_i + \frac{y_i - p_i m_i}{p_i(1-p_i)m_i}, \tag{3}$$

and the diagonal weight matrix W with diagonal w is defined as

$$W_{ii} = w_i = m_i p_i (1-p_i), \tag{4}$$

where we have used the simplifying notation

$$p_i = \frac{1}{1 + exp(-g_i)}. \tag{5}$$

The algorithm, which is outlined in Algorithm 1, consists of an iteratively re-weighted least squares procedure. We iteratively solve the equation

$$(R + \alpha Q^T W^{-1} Q)\gamma = Q^T z, \tag{6}$$

which enables us to obtain a new estimate of g by calculating

$$g = z - \alpha W^{-1} Q\gamma.$$

The new γ and g allows us to reestimate z, W, w, and p before the next iteration.

Every fifth iteration we update our roughness penalty α by minimizing the cross validation error, estimated by

$$CV(\alpha) = \sum_{i=1}^{n} w_i \left(\frac{z_i - g_i}{(1 - A_{ii})^2} \right)^2$$

where

$$A(\alpha) = (I + \alpha Q R^{-1} Q^T)^{-1}.$$

We search for the α that minimizes $CV(\alpha)$ using golden section search [Kiefer, 1953].

Algorithm 1 Pseudocode description of the nonparametric Logistic regression. We are given three vectors of dimension N each representing the total number of scores, m, the median score, $\hat{x}$, and the number of decoy scores y in a set of N bins. We use these vectors to estimate a link function $g(x)$, which we model with a natural cubic spline $\hat{g}(x)$, which take the values g and second derivative γ at the spline knots $\hat{x}$.

procedure ITERATIVE LEAST SQUARE(x, y, m, C, ϵ)
  $g_1, \ldots, g_n \leftarrow (y_1 + C)/(m_1 + 2C), \ldots, (y_n + C)/(m_n + 2C)$ ▷ Initialize g using pseudo counts.
  $\alpha^{new} \leftarrow \alpha^{init}$ ▷ Initiate α to a constant.
  $Q, R \leftarrow$initQR(x) ▷ Initiate Q and R using Eq. 1 and 2.
  repeat
    $\alpha \leftarrow \alpha^{new}$
    for $i \leftarrow 1 \ldots 5$ **do**
      $p, w, z \leftarrow$updateVec(g, m, y) ▷ Update vectors using Eq. 3, 4 and 5
      $\gamma \leftarrow (R + \alpha Q^T W^{-1} Q) \backslash Q^T z$ ▷ Solve Eq. 6
      $g \leftarrow z - \alpha W^{-1} Q\gamma$
    end for
    $\alpha^{new} \leftarrow$linearSearch(CV) ▷ Golden section search
  until $(CV(\alpha) - CV(\alpha^{new}))/CV(\alpha) < \epsilon$
  return (g, γ)
end procedure

References

PJ Green and BW Silverman. *Nonparametric Regression and Generalized Linear Models: A Roughness Penalty Approach*. Chapman & Hall/CRC, 1994.

J. Kiefer. Sequential Minimax Search for a Maximum. *Proceedings of the American Mathematical Society*, 4(3):502–506, 1953.