Solving the weighted linear least squares problem for LLRR and LLR

Petri Koistinen

June 12, 2006

When we calculate the LLRR estimate at evaluation point x we have to solve the linear least squares problem

$$\sum_{i=1}^{k} w_i \left(y_{q(x,i)} - \beta_0 - \beta^T x_{q(x,i)} \right)^2 + \lambda \beta^T \beta = \min_{\beta_0, \beta}! \tag{1}$$

Here $x_{q(x,1)}, \ldots, x_{q(x,k)}$ are the k nearest neighbors (in the feature space) for the evaluation point x; $y_{q(x,1)}, \ldots, y_{q(x,k)}$ are the corresponding (scalar) responses and $w_1, \ldots w_k$ are the weights which depend on the distances of the neareast neighbors from x and also on the weight function used. The weighted linear least squares problem for LLR is otherwise the same, but there $\lambda = 0$. The prediction at x is then

$$\hat{m}(x) = \hat{\beta}_0 + \hat{\beta} x. \tag{2}$$

For a fixed λ and for fixed weights (i.e., for a fixed k and weighting function), the problem (1) can be solved readily in Matlab after one observes that

$$\sum_{i=1}^{k} w_i \left(y_{q(x,i)} - \beta_0 - \beta^T x_{q(x,i)} \right)^2 + \lambda \beta^T \beta = \| A \tilde{\beta} - z \|^2, \tag{3}$$

where

$$A = \begin{bmatrix} \sqrt{w_1} & \sqrt{w_1} x^T_{q(x,1)} \\ \vdots & \vdots \\ \sqrt{w_k} & \sqrt{w_k} x^T_{q(x,k)} \\ \cdots & \cdots \\ 0 & \sqrt{\lambda} I_s \end{bmatrix}, \quad z = \begin{bmatrix} \sqrt{w_1} y_{q(x,1)} \\ \vdots \\ \sqrt{w_k} y_{q(x,k)} \\ \cdots \\ 0_{s \times 1} \end{bmatrix}, \quad \tilde{\beta} = \begin{bmatrix} \beta_0 \\ \beta \end{bmatrix}.$$

Here s is the dimension of x, I_s is the $s \times s$ unit matrix and $0_{s\times 1}$ is the zero vector with s components. Having calculated the matrix A and vector z, the fitted β:s and the prediction at x could be calculated in Matlab as follows

```
beta_aug_fitted = A \ z;
y_pred = beta_aug_fitted' * [1; x];
```

Solving the LLR problem is similar, but there we can omit the lowest block from the matrix A and the vector z.

However, the actual implementations made available here solve the weighted linear least squares problem in a numerically more stable and somewhat more efficient way, which is based on the idea in Seifert & Gasser (2000, Section 2). The crucial observation is that the original weighted least squares problem can be substituted with the following problem

$$\sum_{i=1}^{k} w_i \left(y_{q(x,i)} - \beta_0 - \beta^T (x_{q(x,i)} - u)\right)^2 + \lambda \beta^T \beta = \min_{\beta_0, \beta}!, \tag{4}$$

where u is an arbitrary (centering) vector, provided one uses

$$\hat{m}(x) = \hat{\beta}_0 + \hat{\beta}^T (x - u). \tag{5}$$

instead of (2) to calculate the prediction.

Now the idea is to choose the centering vector u so that the normal equations associated with (4) for the constant β_0 and the slope vector β separate. The normal equations read

$$\begin{bmatrix} \hat{\beta}_0 \\ \hat{\beta} \end{bmatrix} = (A^T A)^{-1} A^T z,$$

where z is as before, but now A is given by

$$A = \begin{bmatrix} \sqrt{w_1} & \sqrt{w_1}(x_{q(x,1)} - u)^T \\ \vdots & \vdots \\ \sqrt{w_k} & \sqrt{w_k}(x_{q(x,k)} - u)^T \\ \cdots & \cdots \\ 0 & \sqrt{\lambda} I_s \end{bmatrix} \tag{6}$$

Hence

$$A^T A = \begin{bmatrix} \sum_{i=1}^k w_i & \sum_{i=1}^k w_i (x_{q(x,i)} - u)^T \\ \sum_{i=1}^k w_i (x_{q(x,i)} - u) & S + \lambda I_s, \end{bmatrix}$$

where S is the matrix

$$S = \sum_{i=1}^{k} w_i (x_{q(x,i)} - u)(x_{q(x,i)} - u)^T. \tag{7}$$

We can arrange the off-diagonal blocks of $A^T A$ to vanish, if we choose u as the weighted average of the nearest neighbors $x_{q(x,i)}$ using weights w_i,

$$u = \frac{\sum_{i=1}^{k} w_i x_{q(x,i)}}{\sum_{j=1}^{k} w_j}. \tag{8}$$

With this choice in (5) and (7) we get, after some algebra,

$$\hat{m}(x) = \frac{\sum_{i=1}^{k} w_i y_{q(x,i)}}{\sum_{j=1}^{k} w_j} + (x-u)^T (S + \lambda I_s)^{-1} \sum_{i=1}^{k} w_i (x_{q(x,i)} - u) y_{q(x,i)}.$$

Rewriting the last formula, we have

$$\hat{m}(x) = \sum_{i=1}^{k} v_i y_{q(x,i)}, \tag{9}$$

where the effective weights v_i are given by

$$v_i = \frac{w_i}{\sum_{j=1}^{k} w_j} + w_i (x-u)^T (S + \lambda I_s)^{-1} (x_{q(x,i)} - u) \tag{10}$$

Since the effective weights do not depend on the y-part of the training data, formula (9) is valid even when the responses y are vectors, and when the same smoothing parameters are used for all the components of y.

References

Seifert B., & Gasser T. (2000), Data adaptive ridging in local polynomial regression. *Journal of Computational and Graphical Statistics*, 9, 338–360.