Fast Leave-one-out Cross-validation for Regularized Least Squares Classification

Jason D. M. Rennie
jrennie@csail.mit.edu

January 15, 2004

Abstract

Unlike other regularized linear classifiers such as the Support Vector Machine and Logistic Regression, parameters for Regularized Least Squares Classification (RLSC) can be learned via a single matrix inverse. This can be extended to calculate outputs for leave-one-out cross-validation using the same matrix inverse. Thus, the regularization parameter can be selected and classification weights for a multiclass problem can be learned using a single matrix inverse. The size of the matrix to be inverted is the lesser of (1) the number of training examples, and (2) the dimensionality of the example space, so selecting a regularization parameter and training RLSC can be highly efficient for problems where either the number of training examples or the dimensionality of the example space is small.

Let X be a matrix of training examples (one per row); let $\mathbf{y}$ be a column vector of their binary ($\{+1, -1\}$) labels. Let $\mathbf{w}$ be the parameter vector for the linear decision boundary to be learned. Using the squared L2-norm for regularization (as is common), Regularized Least Squares Classification (RLSC) [1] minimizes

$$\sum_i (X\mathbf{w} - \mathbf{y})^T (X\mathbf{w} - \mathbf{y}) + \lambda \mathbf{w}^T \mathbf{w}. \tag{1}$$

By setting the derivative with respect to $\mathbf{w}$, we find that we can solve for $\mathbf{w}$ via matrix inverse,

$$\mathbf{w} = (X^T X + \lambda I)^{-1} X^T \mathbf{y} \tag{2}$$

The Representer Theorem ([1], Appendix B) shows that we can safely replace $\mathbf{w}$ with $X^T \mathbf{c}$. This gives an alternate method of solving for the decision boundary parameters:

$$\mathbf{c} = (X X^T + \lambda I)^{-1} \mathbf{y}, \qquad \mathbf{w} = X^T \mathbf{c}. \tag{3}$$

Now we turn to the problem of Leave-one-out cross-validation (LOOCV). LOOCV computes the output for each example using parameters trained on the remaining examples. Naively retraining the classifier for each example can be quite expensive. Here we show that LOOCV outputs can be computed for RLSC using a single matrix inverse[1].

Let $f(\mathbf{x}_j)$ be the RLSC output for example j when RLSC is trained on all examples. Let $f_i(\mathbf{x}_j)$ be the RLSC output for example j when RLSC is trained on all examples except $\mathbf{x}_i$. Let $\mathbf{Y}^i$ be the column vector where $Y^i_j = y_j$ for $j \neq i$ and $Y^i_i = f_i(\mathbf{x}_i)$. Since $f_i(\cdot)$ is the RLSC classifier trained on all examples except $\mathbf{x}_i$, it minimizes

$$\sum_{j \neq i} (Y^i_j - f_i(\mathbf{x}_j))^2 + \lambda \mathbf{w}^T \mathbf{w}. \tag{4}$$

By definition, $f_i(\mathbf{x}_i) = Y^i_i$, so it also minimizes

$$\sum_{j} (Y^i_j - f_i(\mathbf{x}_j))^2 + \lambda \mathbf{w}^T \mathbf{w}, \tag{5}$$

where the sum is now over all training examples. Thus, f_i is the solution to an RLSC task where the training example "labels" are $\mathbf{Y}_i$. Let $H = (X^T X + \lambda I)^{-1}$. Then,

$$f_i(\mathbf{x_i}) = \sum_j (XHX^T)_{ij} Y^i_j, \tag{6}$$

$$= f(\mathbf{x}_i) - (XHX^T)_{ii} y_i + (XHX^T)_{ii} f_i(\mathbf{x}_i), \tag{7}$$

$$= \frac{f(\mathbf{x}_i) - (XHX^T)_{ii} y_i}{1 - (XHX^T)_{ii}}. \tag{8}$$

The last step gives the computation for LOOCV outputs. Let $G = (XX^T + \lambda I)^{-1}$. Substituting $\mathbf{w} = X^T \mathbf{c}$, a similar line of reasoning gives us,

$$f_i(\mathbf{x_i}) = \frac{f(\mathbf{x}_i) - (XX^T G)_{ii} y_i}{1 - (XX^T G)_{ii}}. \tag{9}$$

References

[1] Ryan Rifkin. *Everything Old Is New Again: A Fresh Look at Historical Approaches in Machine Learning*. PhD thesis, Massachusetts Institute of Technology, 2002.

[1] This proof can also be found in [1]