Notes on recursive least squares

Michael I. Jordan
University of California, Berkeley
10/26/98

Scalar output

Consider a set of observations $\{\mathbf{x}_i, y_i\}$, where $\mathbf{x}_i$ is a vector input and y_i is a scalar output. The regression model is:

$$y_i = \mathbf{x}_i^T \boldsymbol{\theta},$$

where $\boldsymbol{\theta}$ is the parameter vector. The least squares problem involves minimizing the following cost function:

$$J = \frac{1}{2} \sum_{i=1}^{n} (y_i - \mathbf{x}_i^T \boldsymbol{\theta}).$$

Let X_n denote the regression matrix based on n observations. That is, X_n has n rows, which are the input vectors $\mathbf{x}_i^T$, $i = 1, 2, \cdots, n$. Let R_n denote the matrix of sums of squares and cross-products of the n input vectors:

$$R_n = X_n^T X_n.$$

Recursive least squares is composed of two equations. The first is a recursive update for the parameter vector. Let $\hat{\boldsymbol{\theta}}_n$ denote the estimated parameter vector at the n^{th} time step (i.e., after n observations have arrived). The estimate is updated recursively as follows:

$$\hat{\boldsymbol{\theta}}_n = \hat{\boldsymbol{\theta}}_{n-1} + R_n^{-1} \mathbf{x}_n (y_n - \mathbf{x}_n^T \hat{\boldsymbol{\theta}}_{n-1}).$$

The covariance matrix R_n^{-1} is also updated recursively, as follows:

$$R_n^{-1} = R_{n-1}^{-1} - R_{n-1}^{-1} \mathbf{x}_n (1 + \mathbf{x}_n^T R_{n-1}^{-1} \mathbf{x}_n)^{-1} \mathbf{x}_n^T R_{n-1}^{-1}.$$

(Note that the term in the parentheses is a scalar. No matrices need to be inverted). The recursion is started with an initial value for R_0^{-1}. A typical choice is a diagonal matrix with large entries on its diagonal. This corresponds to large uncertainty about the initial values of the parameters (priors with high variance).

Proof

Consider partitioned forms of X_n and $\mathbf{y}_n$:

$$X_n = \begin{bmatrix} X_{n-1} \\ \mathbf{x}_n^T \end{bmatrix}, \quad \mathbf{y}_n = \begin{bmatrix} \mathbf{y}_{n-1} \\ y_n \end{bmatrix}.$$

The partioned form of X_n allows R_n to be decomposed as:

$$R_n = R_{n-1} + \mathbf{x}_n \mathbf{x}_n^T.$$

Recall that the normal equations provide necessary and sufficient conditions for the optimal parameters $\hat{\boldsymbol{\theta}}$:

$$R_n \hat{\boldsymbol{\theta}}_n = X_n^T \mathbf{y}_n.$$

Expand the normal equations as follows:

$$\begin{aligned} R_n \hat{\boldsymbol{\theta}}_n &= X_n^T \mathbf{y}_n \\ &= X_{n-1}^T \mathbf{y}_{n-1} + \mathbf{x}_n y_n \\ &= R_{n-1} \hat{\boldsymbol{\theta}}_{n-1} + \mathbf{x}_n y_n \\ &= (R_n - \mathbf{x}_n \mathbf{x}_n^T) \hat{\boldsymbol{\theta}}_{n-1} + \mathbf{x}_n y_n \\ &= R_n \hat{\boldsymbol{\theta}}_{n-1} + \mathbf{x}_n (y_n - \mathbf{x}_n^T \hat{\boldsymbol{\theta}}_{n-1}), \end{aligned}$$

which implies that

$$\hat{\boldsymbol{\theta}}_n = \hat{\boldsymbol{\theta}}_{n-1} + R_n^{-1} \mathbf{x}_n (y_n - \mathbf{x}_n^T \hat{\boldsymbol{\theta}}_{n-1}).$$

The update for the covariance matrix can be obtained from the matrix inversion lemma:

$$(A + BCD)^{-1} = A^{-1} - A^{-1} B (C^{-1} + DA^{-1}B) DA^{-1},$$

substituting $A = R_{n-1}$, $B = \mathbf{x}_n$, $C = 1$, and $D = \mathbf{x}_n^T$.

Recursive weighted least squares

We include a diagonal weighting matrix W in the least squares problem. The normal equations for weighted least squares are:

$$X^T W X \hat{\boldsymbol{\theta}} = X^T W \mathbf{y}.$$

The parameter update equation for recursive weighted least squares is as follows:

$$\hat{\boldsymbol{\theta}}_n = \hat{\boldsymbol{\theta}}_{n-1} + R_n^{-1} \mathbf{x}_n w_n (y_n - \mathbf{x}_n^T \hat{\boldsymbol{\theta}}_{n-1}),$$

and the covariance matrix update equation is:

$$R_n^{-1} = R_{n-1}^{-1} - R_{n-1}^{-1} \mathbf{x}_n (w_n^{-1} + \mathbf{x}_n^T R_{n-1}^{-1} \mathbf{x}_n)^{-1} \mathbf{x}_n^T R_{n-1}^{-1},$$

where w_n is the n^{th} diagonal element of W.

Proof

Define R_n as the matrix of weighted sums of squares and cross-products:

$$\begin{aligned} R_n &= X_n^T W_n X_n \\ &= R_{n-1} + \mathbf{x}_n w_n \mathbf{x}_n^T, \end{aligned}$$

where $W_n = \mathrm{diag}(w_1, w_2, \ldots, w_n)$. The proof is essentially the same as the unweighted case:

$$\begin{aligned} R_n\hat{\boldsymbol{\theta}}_n &= X_n^T W_n \mathbf{y}_n \\ &= X_{n-1}^T W_{n-1}\mathbf{y}_{n-1} + \mathbf{x}_n w_n y_n \\ &= R_{n-1}\hat{\boldsymbol{\theta}}_{n-1} + \mathbf{x}_n w_n y_n \\ &= (R_n - \mathbf{x}_n w_n \mathbf{x}_n^T)\hat{\boldsymbol{\theta}}_{n-1} + \mathbf{x}_n w_n y_n \\ &= R_n\hat{\boldsymbol{\theta}}_{n-1} + \mathbf{x}_n w_n (y_n - \mathbf{x}_n^T\hat{\boldsymbol{\theta}}_{n-1}), \end{aligned}$$

which implies that

$$\hat{\boldsymbol{\theta}}_n = \hat{\boldsymbol{\theta}}_{n-1} + R_n^{-1}\mathbf{x}_n w_n (y_n - \mathbf{x}_n^T\hat{\boldsymbol{\theta}}_{n-1}).$$

The covariance matrix update can be obtained by substituting $C = w_n$ in the matrix inversion lemma.

Vector output

The vector output case can be treated as a set of scalar output cases, and the analysis for the scalar case goes through mutatis mutandi.

The regression model is:

$$\mathbf{y}_i^T = \mathbf{x}_i^T \boldsymbol{\Theta},$$

where $\mathbf{x}_i$ is the i^{th} input vector, $\mathbf{y}_i$ is the i^{th} output vector, and $\boldsymbol{\Theta}$ is the parameter matrix. Letting X and Y denote the matrices whose rows are $\mathbf{x}_i^T$ and $\mathbf{y}_i^T$, respectively, the normal equations are:

$$X^T X\hat{\boldsymbol{\Theta}} = X^T Y,$$

which can be thought of as a set of scalar output problems.

The parameter matrix is updated recursively as follows:

$$\hat{\boldsymbol{\Theta}}_n = \hat{\boldsymbol{\Theta}}_{n-1} + R_n^{-1}\mathbf{x}_n(\mathbf{y}_n^T - \mathbf{x}_n^T\hat{\boldsymbol{\Theta}}_{n-1}).$$

The update for the covariance matrix R_n^{-1} is exactly the same as in the scalar case:

$$R_n^{-1} = R_{n-1}^{-1} - R_{n-1}^{-1}\mathbf{x}_n(1 + \mathbf{x}_n^T R_{n-1}^{-1}\mathbf{x}_n)^{-1}\mathbf{x}_n^T R_{n-1}^{-1}.$$

Note that no matrices have to be inverted in either update equation.

Recursive least squares with exponential forgetting

Recursive least squares with exponential forgetting is based on the following cost function:

$$J = \frac{1}{2}\sum_{i=1}^{n} \lambda^{n-i}(y_i - \mathbf{x}_i^T\boldsymbol{\theta}),$$

where λ is a discount parameter.

The parameter update equation for recursive least squares with exponential forgetting is as follows:

$$\hat{\boldsymbol{\theta}}_n = \hat{\boldsymbol{\theta}}_{n-1} + R_n^{-1}\mathbf{x}_n(y_n - \mathbf{x}_n^T\hat{\boldsymbol{\theta}}_{n-1}),$$

and the covariance matrix update equation is:

$$R_n^{-1} = \lambda^{-1}[R_{n-1}^{-1} - R_{n-1}^{-1}\mathbf{x}_n(\lambda + \mathbf{x}_n^T R_{n-1}^{-1}\mathbf{x}_n)^{-1}\mathbf{x}_n^T R_{n-1}^{-1}].$$

That is, the parameter update equation is the same as ordinary recursive least squares—only the covariance matrix update equation is different.

Proof

Least squares with exponential forgetting is equivalent to weighted least squares in which the weighting matrix is given by $\Lambda_n = \mathrm{diag}(\lambda^{n-1}, \lambda^{n-2}, \ldots, \lambda)$. The recursive case, however, is not equivalent to recursive weighted least squares, because all of the diagonal entries of Λ change at each time step. Thus:

$$\begin{aligned} \Lambda &= \begin{bmatrix} \lambda\Lambda_{n-1} & 0 \\ 0 & 1 \end{bmatrix}, \\ &= \lambda \begin{bmatrix} \Lambda_{n-1} & 0 \\ 0 & \lambda^{-1} \end{bmatrix}. \end{aligned} \tag{1}$$

Let R_n denote the weighted matrix of sums of squares and cross-products of the n input vectors:

$$R_n = X_n^T\Lambda_n X_n.$$

This matrix can be decomposed as follows:

$$R_n = \lambda R_{n-1} + \mathbf{x}_n\mathbf{x}_n^T.$$

Substituting this relation into the normal equations yields the following expansion:

$$\begin{aligned} R_n\hat{\boldsymbol{\theta}}_n &= \lambda X_n^T\mathbf{y}_n \\ &= \lambda X_{n-1}^T\Lambda_{n-1}\mathbf{y}_{n-1} + \mathbf{x}_n y_n \\ &= \lambda R_{n-1}\hat{\boldsymbol{\theta}}_{n-1} + \mathbf{x}_n y_n \\ &= \lambda(\lambda^{-1}R_n - \lambda^{-1}\mathbf{x}_n\mathbf{x}_n^T)\hat{\boldsymbol{\theta}}_{n-1} + \mathbf{x}_n y_n \\ &= R_n\hat{\boldsymbol{\theta}}_{n-1} + \mathbf{x}_n(y_n - \mathbf{x}_n^T\hat{\boldsymbol{\theta}}_{n-1}), \end{aligned}$$

which implies that

$$\hat{\boldsymbol{\theta}}_n = \hat{\boldsymbol{\theta}}_{n-1} + R_n^{-1}\mathbf{x}_n(y_n - \mathbf{x}_n^T\hat{\boldsymbol{\theta}}_{n-1}).$$

Using the relation $R_n = \lambda R_{n-1} + \mathbf{x}_n\mathbf{x}_n^T$ in the matrix inversion lemma yields the update for the covariance matrix, where $A = \lambda R_{n-1}$, $B = \mathbf{x}_n$, $C = 1$, and $D = \mathbf{x}_n^T$.

Recursive weighted least squares with exponential forgetting

Recursive least squares with exponential forgetting can be combined with recursive weighted least squares in the obvious way:

$$\hat{\boldsymbol{\theta}}_n = \hat{\boldsymbol{\theta}}_{n-1} + R_n^{-1}\mathbf{x}_n w_n(y_n - \mathbf{x}_n^T\hat{\boldsymbol{\theta}}_{n-1}),$$

and:

$$R_n^{-1} = \lambda^{-1}[R_{n-1}^{-1} - R_{n-1}^{-1}\mathbf{x}_n(\lambda w_n^{-1} + \mathbf{x}_n^T R_{n-1}^{-1}\mathbf{x}_n)^{-1}\mathbf{x}_n^T R_{n-1}^{-1}].$$