Notes on Optimisation

G.A. Jarrad

June 24, 2015

1 Introduction

blah, blah, blah

2 Recursive Least Square Error Estimation

Consider using least square error optimisation to approximately fit the linear coefficient model

$$y \quad = \quad \vec{a} \cdot \vec{x} + \varepsilon \tag{2.1}$$

to the data $\{(\vec{x}_t, y_t) \mid t = 1, 2, \ldots, n\}$. The square error is given by

$$S_n \quad = \quad \sum_{t=1}^{n} (y_t - \vec{a} \cdot \vec{x}_t)^2 \tag{2.2}$$

and hence

$$\frac{\partial S_n}{\partial \vec{a}} \quad = \quad -2 \sum_{t=1}^{n} (y_t - \vec{a} \cdot \vec{x}_t) \vec{x}_t \,. \tag{2.3}$$

The turning point $\hat{\vec{a}}_n$ then occurs when

$$\sum_{t=1}^{n} y_t \vec{x}_t = \sum_{t=1}^{n} \vec{x}_t \vec{x}_t^T \hat{\vec{a}}_n \,, \tag{2.4}$$

and thus S_n is maximised when

$$\hat{\vec{a}}_n = \left(\sum_{t=1}^{n} \vec{x}_t \vec{x}_t^T \right)^{-1} \sum_{t=1}^{n} y_t \vec{x}_t \,. \tag{2.5}$$

Suppose now that another data point $(y_{n+1}, \vec{x}_{n+1})$ is given. Then the new least square error estimate is given by

$$\hat{\vec{a}}_{n+1} = \left(\sum_{t=1}^{n+1} \vec{x}_t \vec{x}_t^T \right)^{-1} \sum_{t=1}^{n+1} y_t \vec{x}_t \,, \tag{2.6}$$

which appears to require yet another matrix inverse. However, this secondary inverse can avoided by using recursive estimation of the form

$$\begin{aligned}
P_{n+1} &= \left(\sum_{t=1}^{n+1} \vec{x}_t \vec{x}_t^T\right)^{-1} \\
&= \left(\sum_{t=1}^{n} \vec{x}_t \vec{x}_t^T + \vec{x}_{n+1}\vec{x}_{n+1}^T\right)^{-1} \\
&= \left(P_n^{-1} + \vec{x}_{n+1}\vec{x}_{n+1}^T\right)^{-1} \\
&= P_n - \frac{P_n \vec{x}_{n+1}\vec{x}_{n+1}^T P_n}{1 + \vec{x}_{n+1}^T P_n \vec{x}_{n+1}}\,, \qquad (2.7)
\end{aligned}$$

from the Woodbury matrix identity

$$(A + UCV)^{-1} = A^{-1} - A^{-1}U\left(C^{-1} + VA^{-1}U\right)^{-1} VA^{-1}\,. \qquad (2.8)$$

Consequently, observe that

$$\begin{aligned}
\sum_{t=1}^{n+1} y_t \vec{x}_t &= \sum_{t=1}^{n} y_t \vec{x}_t + y_{n+1}\vec{x}_{n+1} \\
&= \sum_{t=1}^{n} \vec{x}_t \vec{x}_t^T \hat{\vec{a}}_n + y_{n+1}\vec{x}_{n+1} \\
&= \sum_{t=1}^{n+1} \vec{x}_t \vec{x}_t^T \hat{\vec{a}}_n - \vec{x}_{n+1}\vec{x}_{n+1}^T \hat{\vec{a}}_n + y_{n+1}\vec{x}_{n+1} \\
&= P_{n+1}^{-1}\hat{\vec{a}}_n + \vec{x}_{n+1}(y_{n+1} - \hat{\vec{a}}_n \cdot \vec{x}_{n+1}) \qquad (2.9)
\end{aligned}$$

from equation (2.4), and hence the estimate $\hat{\vec{a}}_{n+1}$ is given recursively by

$$\hat{\vec{a}}_{n+1} = \hat{\vec{a}}_n + P_{n+1}\vec{x}_{n+1}(y_{n+1} - \hat{\vec{a}}_n \cdot \vec{x}_{n+1}) \qquad (2.10)$$

from equation (2.6). Observe that this recursive estimate takes the form of a predictor–corrector update, since $\hat{\vec{a}}_n \cdot \vec{x}_{n+1}$ is the expected value of y_{n+1} given $\hat{\vec{a}}_n$, from model (2.1).

As a slight alternative to the above approach to re-estimation, suppose now instead that we are using a sliding data window of fixed width n, e.g. for time series analysis. Then the k-th parameter estimate $\hat{\vec{a}}_n^{(k)}$ is given by

$$\hat{\vec{a}}_n^{(k)} = \left(\sum_{t=k}^{n+k-1} \vec{x}_t \vec{x}_t^T\right)^{-1} \sum_{t=k}^{n+k-1} y_t \vec{x}_t\,. \qquad (2.11)$$

Hence, observe that

$$
\begin{aligned}
P_n^{(k+1)} &= \left(\sum_{t=k+1}^{n+k} \vec{x}_t \vec{x}_t^T\right)^{-1} \\
&= \left(\sum_{t=k}^{n+k-1} \vec{x}_t \vec{x}_t^T - \vec{x}_k \vec{x}_k^T + \vec{x}_{n+k} \vec{x}_{n+k}^T\right)^{-1} \\
&= \left({Q_n^{(k)}}^{-1} + \vec{x}_{n+k} \vec{x}_{n+k}^T\right)^{-1} \\
&= Q_n^{(k)} - \frac{Q_n^{(k)} \vec{x}_{n+k} \vec{x}_{n+k}^T Q_n^{(k)}}{1 + \vec{x}_{n+k}^T Q_n^{(k)} \vec{x}_{n+k}}\,, \qquad (2.12)
\end{aligned}
$$

where

$$
\begin{aligned}
Q_n^{(k)} &= \left(\sum_{t=k}^{n+k-1} \vec{x}_t \vec{x}_t^T - \vec{x}_k \vec{x}_k^T\right)^{-1} \\
&= \left({P_n^{(k)}}^{-1} - \vec{x}_k \vec{x}_k^T\right)^{-1} \\
&= P_n^{(k)} + \frac{P_n^{(k)} \vec{x}_k \vec{x}_k^T P_n^{(k)}}{1 - \vec{x}_k^T P_n^{(k)} \vec{x}_k}\,. \qquad (2.13)
\end{aligned}
$$

Consequently,

$$
\begin{aligned}
\sum_{t=k+1}^{n+k} y_t \vec{x}_t &= \sum_{t=k}^{n+k-1} y_t \vec{x}_t - y_k \vec{x}_k + y_{n+k} \vec{x}_{n+k} \\
&= \sum_{t=k}^{n+k-1} \vec{x}_t \vec{x}_t^T \hat{\vec{a}}_n^{(k)} - y_k \vec{x}_k + y_{n+k} \vec{x}_{n+k} \\
&= \sum_{t=k+1}^{n+k} \vec{x}_t \vec{x}_t^T \hat{\vec{a}}_n^{(k)} + \vec{x}_k \vec{x}_k^T \hat{\vec{a}}_n^{(k)} - \vec{x}_{n+k} \vec{x}_{n+k}^T \hat{\vec{a}}_n^{(k)} \\
&\qquad - y_k \vec{x}_k + y_{n+k} \vec{x}_{n+k} \\
&= {P_n^{(k+1)}}^{-1} \hat{\vec{a}}_n^{(k)} - \vec{x}_k (y_k - \hat{\vec{a}}_n^{(k)} \cdot \vec{x}_k) \\
&\qquad + \vec{x}_{n+k} (y_{n+k} - \hat{\vec{a}}_n^{(k)} \cdot \vec{x}_{n+k})\,, \qquad (2.14)
\end{aligned}
$$

and hence

$$
\begin{aligned}
\hat{\vec{a}}_n^{(k+1)} &= \hat{\vec{a}}_n^{(k)} - P_n^{(k+1)} \vec{x}_k (y_k - \hat{\vec{a}}_n^{(k)} \cdot \vec{x}_k) \\
&\qquad + P_n^{(k+1)} \vec{x}_{n+k} (y_{n+k} - \hat{\vec{a}}_n^{(k)} \cdot \vec{x}_{n+k})\,, \qquad (2.15)
\end{aligned}
$$

from equation (2.11).