LUND UNIVERSITY

Recursive least squares

Andreas Jakobsson

Recursive least squares

Recall the linear model discussed earlier

$$y_t = \mathbf{x}_t^T \boldsymbol{\theta} + e_t$$

where e_t is an additive noise process and the regressor $\mathbf{x}_t$ is *known* vector, wheras $\boldsymbol{\theta}$ is an $n_{\boldsymbol{\theta}}$-dimensional vector containing the parameters of interest. In our earlier discussion, we considered the estimation of $\boldsymbol{\theta}$ using N samples.

Let

$$\begin{aligned} \mathbf{y}_N &= \begin{bmatrix} y_1 & \dots & y_N \end{bmatrix}^T \\ \mathbf{X}_N &= \begin{bmatrix} \mathbf{x}_1 & \dots & \mathbf{x}_N \end{bmatrix}^* \\ \mathbf{e}_N &= \begin{bmatrix} e_1 & \dots & e_N \end{bmatrix}^T \end{aligned}$$

Then, as seen earlier, the least squares (LS) estimate of $\boldsymbol{\theta}$ can be found as

$$\hat{\boldsymbol{\theta}}_{LS} = (\mathbf{X}_N^* \mathbf{X}_N)^{-1} \mathbf{X}_N^* \mathbf{y}_N$$

Recursive least squares

We are now interested in the case when one wish to recompute the estimate $\hat{\boldsymbol{\theta}}_{LS}$ as additional data becomes available. One may then of course simply recompute the LS estimate as above, but this is computationally inefficient.

Lets consider the estimate computed at time t, $\hat{\boldsymbol{\theta}}_t$, and introduce

$$\mathbf{R}_t = \sum_{\ell=p+1}^{t} \mathbf{x}_\ell \mathbf{x}_\ell^T$$

$$\mathbf{g}_t = \sum_{\ell=p+1}^{t} \mathbf{x}_\ell y_\ell$$

such that

$$\hat{\boldsymbol{\theta}}_t = (\mathbf{X}_t^* \mathbf{X}_t)^{-1} \mathbf{X}_t^* \mathbf{y}_t = \mathbf{R}_t^{-1} \mathbf{g}_t$$

Recursive least squares

We are now interested in the case when one wish to recompute the estimate $\hat{\boldsymbol{\theta}}_{LS}$ as additional data becomes available. One may then of course simply recompute the LS estimate as above, but this is computationally inefficient.

Lets consider the estimate computed at time t, $\hat{\boldsymbol{\theta}}_t$, and introduce

$$\mathbf{R}_t = \sum_{\ell=p+1}^{t} \mathbf{x}_\ell \mathbf{x}_\ell^T$$

$$\mathbf{g}_t = \sum_{\ell=p+1}^{t} \mathbf{x}_\ell y_\ell$$

such that

$$\hat{\boldsymbol{\theta}}_t = (\mathbf{X}_t^* \mathbf{X}_t)^{-1} \mathbf{X}_t^* \mathbf{y}_t = \mathbf{R}_t^{-1} \mathbf{g}_t$$

Lets now divide the sums into the "earlier" contribution, i.e., the one using data up to $t-1$, and the most recent sample, at time t, i.e.,

$$\mathbf{R}_t = \mathbf{R}_{t-1} + \mathbf{x}_t \mathbf{x}_t^T$$

$$\mathbf{g}_t = \mathbf{g}_{t-1} + \mathbf{x}_t y_t$$

This means that

$$\begin{aligned}
\hat{\boldsymbol{\theta}}_t &= \mathbf{R}_t^{-1} \left[\mathbf{g}_{t-1} + \mathbf{x}_t y_t \right] \\
&= \mathbf{R}_t^{-1} \left[\mathbf{R}_{t-1} \hat{\boldsymbol{\theta}}_{t-1} + \mathbf{x}_t y_t \right] \\
&= \mathbf{R}_t^{-1} \left[\mathbf{R}_t \hat{\boldsymbol{\theta}}_{t-1} - \mathbf{x}_t \mathbf{x}_t^T \hat{\boldsymbol{\theta}}_{t-1} + \mathbf{x}_t y_t \right] \\
&= \hat{\boldsymbol{\theta}}_{t-1} + \mathbf{R}_t^{-1} \mathbf{x}_t \left[y_t - \mathbf{x}_t^T \hat{\boldsymbol{\theta}}_{t-1} \right]
\end{aligned}$$

Recursive least squares

Almost there! However, we do not want to compute the inverse $\mathbf{R}_t^{-1}$ at each update! Recall that $\mathbf{R}_t = \mathbf{R}_{t-1} + \mathbf{x}_t\mathbf{x}_t^T$. The *matrix inversion lemma* states that

$$\left(\mathbf{A} - \mathbf{C}\mathbf{B}^{-1}\mathbf{D}\right)^{-1} = \mathbf{A}^{-1} + \mathbf{A}^{-1}\mathbf{C}\left(\mathbf{B} - \mathbf{D}\mathbf{A}^{-1}\mathbf{C}\right)^{-1}\mathbf{D}\mathbf{A}^{-1}$$

Using $\mathbf{A} = \mathbf{R}_{t-1}$, $\mathbf{C} = \mathbf{D}^T = \mathbf{x}_t$, $\mathbf{B} = -\mathbf{I}$, and defining $\mathbf{P}_t = \mathbf{R}_t^{-1}$, one obtains

$$\begin{aligned}\mathbf{P}_t &= \mathbf{P}_{t-1} + \mathbf{P}_{t-1}\mathbf{x}_t\left(-\mathbf{I} - \mathbf{x}_t^T\mathbf{P}_{t-1}\mathbf{x}_t\right)^{-1}\mathbf{x}_t^T\mathbf{P}_{t-1}\\ &= \mathbf{P}_{t-1} - \frac{\mathbf{P}_{t-1}\mathbf{x}_t\mathbf{x}_t^T\mathbf{P}_{t-1}}{1 + \mathbf{x}_t^T\mathbf{P}_{t-1}\mathbf{x}_t}\end{aligned}$$

Recursive least squares

Almost there! However, we do not want to compute the inverse $\mathbf{R}_t^{-1}$ at each update! Recall that $\mathbf{R}_t = \mathbf{R}_{t-1} + \mathbf{x}_t \mathbf{x}_t^T$. The *matrix inversion lemma* states that

$$\left(\mathbf{A} - \mathbf{C}\mathbf{B}^{-1}\mathbf{D}\right)^{-1} = \mathbf{A}^{-1} + \mathbf{A}^{-1}\mathbf{C}\left(\mathbf{B} - \mathbf{D}\mathbf{A}^{-1}\mathbf{C}\right)^{-1}\mathbf{D}\mathbf{A}^{-1}$$

Using $\mathbf{A} = \mathbf{R}_{t-1}$, $\mathbf{C} = \mathbf{D}^T = \mathbf{x}_t$, $\mathbf{B} = -\mathbf{I}$, and defining $\mathbf{P}_t = \mathbf{R}_t^{-1}$, one obtains

$$\begin{aligned} \mathbf{P}_t &= \mathbf{P}_{t-1} + \mathbf{P}_{t-1}\mathbf{x}_t\left(-\mathbf{I} - \mathbf{x}_t^T\mathbf{P}_{t-1}\mathbf{x}_t\right)^{-1}\mathbf{x}_t^T\mathbf{P}_{t-1} \\ &= \mathbf{P}_{t-1} - \frac{\mathbf{P}_{t-1}\mathbf{x}_t\mathbf{x}_t^T\mathbf{P}_{t-1}}{1 + \mathbf{x}_t^T\mathbf{P}_{t-1}\mathbf{x}_t} \end{aligned}$$

Let the Kalman gain $\mathbf{K}_t = \mathbf{R}_t^{-1}\mathbf{x}_t$, such that

$$\mathbf{K}_t = \mathbf{R}_t^{-1}\mathbf{x}_t = \mathbf{P}_{t-1}\mathbf{x}_t - \frac{\mathbf{P}_{t-1}\mathbf{x}_t\mathbf{x}_t^T\mathbf{P}_{t-1}}{1 + \mathbf{x}_t^T\mathbf{P}_{t-1}\mathbf{x}_t}\mathbf{x}_t = \frac{\mathbf{P}_{t-1}\mathbf{x}_t}{1 + \mathbf{x}_t^T\mathbf{P}_{t-1}\mathbf{x}_t}$$

yields the recursive least squares (RLS) estimate

$$\hat{\boldsymbol{\theta}}_t = \hat{\boldsymbol{\theta}}_{t-1} + \mathbf{K}_t\left[y_t - \mathbf{x}_t^T\hat{\boldsymbol{\theta}}_{t-1}\right]$$

Recursive least squares

It is important to note that using the described implementation the LS and the RLS estimates are *the same*. The RLS is just a recursive implementation of LS.

However, when referring to RLS, one is more commonly referring to a version implemented using a *forgetting factor*, such that older samples influence the estimate less than more current samples. In order to implement this, one instead forms the minimisation from

$$\hat{\boldsymbol{\theta}}_t = \arg\min_{\boldsymbol{\theta}} \sum_{\ell=p+1}^{t} \lambda^{t-\ell} \left[y_\ell - \mathbf{x}_\ell^T \boldsymbol{\theta} \right]^2$$

where $0 < \lambda \leq 1$ is the forgetting factor.

Recursive least squares

It is important to note that using the described implementation the LS and the RLS estimates are *the same*. The RLS is just a recursive implementation of LS.

However, when referring to RLS, one is more commonly referring to a version implemented using a *forgetting factor*, such that older samples influence the estimate less than more current samples. In order to implement this, one instead forms the minimisation from

$$\hat{\boldsymbol{\theta}}_t = \arg\min_{\boldsymbol{\theta}} \sum_{\ell=p+1}^{t} \lambda^{t-\ell} \left[y_\ell - \mathbf{x}_\ell^T \boldsymbol{\theta} \right]^2$$

where $0 < \lambda \leq 1$ is the forgetting factor. The derivation follows the above closely and results in

$$\begin{aligned}
\hat{\boldsymbol{\theta}}_t &= \hat{\boldsymbol{\theta}}_{t-1} + \mathbf{K}_t \left[y_t - \mathbf{x}_t^T \hat{\boldsymbol{\theta}}_{t-1} \right] \\
\mathbf{M}_t &= \mathbf{P}_{t-1} \mathbf{x}_t \\
\mathbf{K}_t &= \frac{\mathbf{M}_t}{\lambda + \mathbf{x}_t^T \mathbf{M}_t} \\
\mathbf{P}_t &= \frac{1}{\lambda} \left(\mathbf{P}_{t-1} - \frac{\mathbf{M}_t \mathbf{M}_t^T}{\lambda + \mathbf{x}_t^T \mathbf{M}_t} \right)
\end{aligned}$$