# Recursive estimation

Erik Lindström

Centre for Mathematical Sciences
Lund University

LU/LTH & DTU

# Overview


Introduction

Naive recursive estimators

Recursive LS

Recursive Pseudo-Linear Regression

Recursive Prediction Error Method

Recursive Maximum Likelihood

Filtering

# Different types

- Forgetting type estimators
- Converging estimators

Ex: $Z_i \in \mathcal{N}(\mu, 1)$. Estimate the mean ($\mu$) as

$$\hat{\mu}_N = \frac{1}{N}\sum Z_i$$

or as

$$\hat{\mu}_N = Z_N?$$

Different properties and applications!

# Naive approaches

- Windowed estimation
- Use $[t-u:t]$ to estimate parameters

$$\hat{\theta}_t = \text{argmax} \sum_{n=t-u}^{t} \log p(y_n | y_{t-u}, \ldots y_{n-1})$$

- Followed by

$$\hat{\theta}_{t+1} = \text{argmax} \sum_{n=t-u+1}^{t+1} \log p(y_n | y_{t-u+1}, \ldots y_{n-1})$$

Properties?

# Recursive LS

- Linear models can be written as

$$Y = X\theta + e$$

- Estimate is given by

$$\hat{\theta} = (X^T X)^{-1}(X^T Y)$$

Can be written in recursive form!

# Recursive LS

- Optimize

$$\hat{\theta}_t = \operatorname{argmin} \sum_{s=p}^{t} (Y_s - X_s^T \theta)^2$$

- where

$$X_t^T = [-Y_{t-1}, \ldots, -Y_{t-p}]$$

and

$$\theta^T = [\theta_1, \ldots, \theta_p]$$

- This can be written as

$$\begin{aligned} \hat{\theta}_t &= R_t^{-1} h_t \\ R_t &= \sum X_s X_s^T \\ h_t &= \sum X_s Y_s \end{aligned} \tag{1}$$

# Recursive LS

- We can now write $R_t = R_{t-1} + X_t X_t^T$
- and $h_t = h_{t-1} + X_t Y_t$

and also

$$\begin{aligned}\hat{\theta}_t &= R_t^{-1} h_t \\ &= R_t^{-1}(h_{t-1} + X_t Y_t) \\ &= R_t^{-1}(R_{t-1}\hat{\theta}_{t-1} + X_t Y_t) \\ &= R_t^{-1}(R_t\hat{\theta}_{t-1} - X_t X_t^T \hat{\theta}_{t-1} + X_t Y_t) \\ &= \hat{\theta}_{t-1} + R_t^{-1} X_t (Y_t - X_t^T \hat{\theta}_{t-1})\end{aligned} \tag{2}$$

This is the standard Recursive LS (RLS)

# Recursive LS

- We have that $R_t = R_{t-1} + X_t X_t^T$
- but are interested in $R_t^{-1}$

The matrix inversion lemma

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

gives

$$R_t^{-1} = R_{t-1}^{-1} - R_{t-1}^{-1}X_t(X_t^T R_{t-1}^{-1}X_t + I)^{-1}X_t^T R_{t-1}^{-1}$$

The RLS algorithm is then given by two simple matrix expressions!

# Adaptive Recursive LS

Optimize

$$\hat{\theta}_t = \operatorname{argmin} \sum_{s=p}^{t} \beta(t,s)(Y_s - X_s^T\theta)^2$$

where

$$\begin{aligned} \beta(t,s) &= \lambda(t)\beta(t-1,s) \\ \beta(t,t) &= 1 \end{aligned} \tag{3}$$

Hence is $\beta(t,s) = \prod_{j=s+1}^{t} \lambda(j)$.
Again, recursive equations can be found!

# Adaptive Recursive LS

The solution is given by

$$\hat{\theta}_t = \tilde{R}_t^{-1}\tilde{h}_t$$

where

- $\tilde{R}_t = \lambda(t)\tilde{R}_{t-1} + X_t X_t^T$
- $\tilde{h}_t = \lambda(t)\tilde{h}_{t-1} + X_t Y_t$

And the rest is identical to the standard RLS.

- Interpretation of $\lambda$.

# Recursive Pseudo-Linear Regression

- Extend

$$Y = X\theta + e$$

- To

$$Y = X(\theta)\theta + e$$

Includes e.g. ARMA and non-linear models!

# (Adaptive) RPLR

Let

$$\hat{\theta}_t = \text{argmin}\, S_t(\theta)$$

where

$$S_t(\theta) = \sum_s^t \beta(t,s)(Y_s - X_s^T(\theta)\theta)^2$$

- $S_t(\theta) = \lambda(t) S_{t-1}(\theta) + (Y_t - X_t^T(\theta)\theta)^2$
- Taylor expand around $\hat{\theta}_{t-1}$

# (Adaptive) RPLR

- Taylor expansion

$$S_t(\theta) \approx S_t(\hat{\theta}_{t-1}) + \nabla S_t(\hat{\theta}_{t-1})(\theta - \hat{\theta}_{t-1}) + \frac{1}{2}(\theta - \hat{\theta}_{t-1})^T H_t(\hat{\theta}_{t-1})(\theta - \hat{\theta}_{t-1}), \quad (4)$$

where $H_t$ is the Hessian.

- $\nabla S_t(\hat{\theta}_{t-1}) \approx -2X_t(Y_t - X_t^T \hat{\theta}_{t-1})$
- $R_t = \frac{1}{2} H_t = \lambda(t) R_{t-1} + X_t X_t^T$
- This gives the estimators as

$$\hat{\theta}_t = \hat{\theta}_{t-1} + R_t^{-1} X_t (Y_t - X_t^T \hat{\theta}_{t-1})$$

# (Adaptive) RPEM

Let

$$\hat{\theta}_t = \text{argmin}\, S_t(\theta)$$

where

$$S_t(\theta) = \sum_s^t \beta(t,s)(Y_s - \hat{Y}_{s|s-1}(\theta))^2$$

- Approximate by a $2nd$ order polynomial
- Optimize using Newton-Raphson

# (Adaptive) RPEM

- Taylor expansion

$$S_t(\theta) \approx S_t(\hat{\theta}_{t-1}) + \nabla S_t(\hat{\theta}_{t-1})(\theta - \hat{\theta}_{t-1}) + \frac{1}{2}(\theta - \hat{\theta}_{t-1})^T H_t(\hat{\theta}_{t-1})(\theta - \hat{\theta}_{t-1}), \tag{5}$$

  where $H_t$ is the Hessian.

- Solution is given by

$$\hat{\theta}_t = \hat{\theta}_{t-1} - H_t(\hat{\theta}_{t-1}) \nabla S_t(\hat{\theta}_{t-1})$$

# (Adaptive) RPEM

Note that

- $S_t(\theta) = \lambda(t)S_{t-1}(\theta) + (Y_t - \hat{Y}_{t|t-1}(\theta))^2$
- $\nabla S_t(\theta) = \lambda(t)\nabla S_{t-1}(\theta) + (Y_t - \hat{Y}_{t|t-1}(\theta))\nabla \hat{Y}_{t|t-1}(\theta)$
- $\nabla S_t(\hat{\theta}_{t-1}) \approx (Y_t - \hat{Y}_{t|t-1}(\hat{\theta}_{t-1}))\nabla \hat{Y}_{t|t-1}(\hat{\theta}_{t-1})$
- The Hessian is given by

$$
\begin{aligned}
H_t(\theta) =& 2\sum \beta(t,s)\nabla \hat{Y}_{s|s-1}(\theta)\nabla \hat{Y}_{s|s-1}^T(\theta) \\
&-2\sum \beta(t,s)\nabla\nabla \hat{Y}_{s|s-1}(\theta)(Y_s - \hat{Y}_{s|s-1}(\theta))
\end{aligned} \tag{6}
$$

- $H_t(\hat{\theta}_{t-1}) \approx \lambda(t)H_{t-1} + 2\nabla \hat{Y}_{t|t-1}(\hat{\theta}_{t-1})\nabla \hat{Y}_{t|t-1}^T(\hat{\theta}_{t-1})$

# (Adaptive) RPEM

This gives

- $R_t = \frac{1}{2} H_t$
- $\hat{\theta}_t = \hat{\theta}_{t-1} + R_t(\hat{\theta}_{t-1})(Y_t - \hat{Y}_{t|t-1}(\hat{\theta}_{t-1}))\nabla \hat{Y}_{t|t-1}(\hat{\theta}_{t-1})$
- $R_t = \lambda(t) R_{t-1} + \nabla \hat{Y}_{t|t-1}(\hat{\theta}_{t-1}) \nabla \hat{Y}_{t|t-1}^T(\hat{\theta}_{t-1})$

Use matrix inversion lemma to obtain an efficient recursion.

# Recursive ML

It is possible to construct recursive estimators for non-Gaussian models

- $\hat{\theta}_t = \text{argmax} \sum_{n=1}^{t} \log p(y_n | y_{1:n-1}, \theta) = \text{argmax}\, \ell_t(\theta)$

Taylor expand and maximize

$$\nabla \ell_t(\hat{\theta}_t) \approx \nabla \ell_t(\hat{\theta}_{t-1}) + \nabla\nabla \ell_t(\hat{\theta}_{t-1})(\hat{\theta}_t - \hat{\theta}_{t-1}) \tag{7}$$

$$= \nabla \ell_{t-1}(\hat{\theta}_{t-1}) + \nabla \log p(y_n | y_{1:n-1}, \hat{\theta}_{t-1}) \tag{8}$$

$$+ \nabla\nabla \ell_t(\hat{\theta}_{t-1})(\hat{\theta}_t - \hat{\theta}_{t-1}) = 0. \tag{9}$$

Simplification gives

$$\hat{\theta}_t = \hat{\theta}_{t-1} + \frac{1}{t} I(\hat{\theta}_{t-1})^{-1} \nabla \log p(y_n | y_{1:n-1}, \hat{\theta}_{t-1})$$

# Robbins-Monro stochastic approximation

- This is a special case of the Robbins-Monro stochastic approximation algorithm
- Problem: $x^\star = \text{argmin}\, G(x)$
- Introduce $x_{n+1} = x_n + \frac{a}{(1+n+A)^\alpha} g(x_n)$
- where $x$ is a parameter, $a$ some positive def. matrix, $g(x)$ is a noisy gradient of $G$ and $\alpha \in (.5, 1]$.
- It then holds that

$$x_n \overset{a.s.}{\to} x^\star \tag{10}$$

$$N^{\alpha/2}(x_n - x^\star) \overset{d}{\to} N(0, \Sigma) \tag{11}$$

Interpretations

# SP/FD stochastic approximation

- The gradient can be approximated by finite difference at the cost of slower convergence.
- but clever methods (SPSA) is still fairly fast
- Idea: Many steps are taken, and the gradient is being averaged over the iterations.
- SPSA only evaluates a single central finite difference (randomly selected) per iteration and averages again over the iterations.

Result: Computational gain is asymp. equal to the dimension of $x$ (which can be huge).

# Filtering

- Recursive estimation using non-linear filters
- Augment

$$x_{n+1} = f(x_n) + e_{n+1} \quad (12)$$
$$y_{n+1} = h(x_{n+1}) + w_{n+1} \quad (13)$$

- to

$$\begin{pmatrix} x_{n+1} \\ \theta_{n+1} \end{pmatrix} = \begin{pmatrix} f(x_n) \\ \theta_n \end{pmatrix} + \begin{pmatrix} e^x_{n+1} \\ e^\theta_{n+1} \end{pmatrix} \quad (14)$$
$$y_{n+1} = h(x_{n+1}, \theta_{n+1}) + w_{n+1} \quad (15)$$

Estimation "trivial", cf. computer exercise 2 and slides on stoch approx.

# Consistent estimates in the filtering setup

- Estimate is often biased.
- Idea. Let $Var[e^{\theta}] \to 0$
- Formalized in the 'iterated filtering framework'
- Can show consistency $\theta_{n+1} \to \theta_0$