SGD for Correlated Data

Moyan Li

September 2019

1 Algorithm

1.1 Parameter Estimation

SGD is a widely used method to find the minimum of a function f. In this project, we try to use SGD in Gaussian Process to optimize the parameters and then make predictions. I will use the RBF kernel as an example to illustrate the algorithm, i.e.

$$K(\mathbf{x_p}, \mathbf{x_q}) = l_1^2 \cdot e^{-\frac{\|\mathbf{x_p} - \mathbf{x_q}\|^2}{2 \cdot l_2^2}}$$

where l_1 and l_2 are parameters to be estimated.

Data set is $(X, \mathbf{y})$, where X is a $n \cdot p$ matrix and $\mathbf{y}$ is a vector. Here n is the number of data and p is the dimension of data.

1.Generate the covariance matrix of $\mathbf{y}$: using $K(\mathbf{x_p}, \mathbf{x_q})$, we could calculate the covariance matrix of $\mathbf{y}$:

$$\text{cov}(y_p, y_q) = K(\mathbf{x_p}, \mathbf{x_q}) + \sigma^2 \delta_{pq} \quad \text{or} \quad \text{cov}(\mathbf{y}) = K(X, X) + \sigma^2 I$$

where δ_{pq} is a Kronecker delta which is one iff $p = q$ and zero otherwise. It follows from the independence assumption about the noise, that a diagonal matrix is added.

2.Generate $\mathbf{y}$: Based on our assumption that $\mathbf{y}|X$ is a Gaussian Process, we could generate $\mathbf{y}$ from $\mathcal{N}(\mu, \Sigma)$, where$\Sigma = K(X, X) + \sigma_n^2 I$ and without loss of generality, we assume $\mu = \mathbf{0}$ in our following discussion

3. Define the negative log-likelihood of $\mathbf{y}$, which is also the objective function in this problem:

$$\begin{aligned} L(\Theta, data) &= -log[(2\pi)^{-2/n} |\Sigma|^{-n} e^{-\frac{1}{2}((\mathbf{y}-\mu)^T \Sigma^{-1} (\mathbf{y}-\mu))}] \\ &= \tfrac{1}{2}((\mathbf{y} - \mu)^T \Sigma^{-1} (\mathbf{y} - \mu) + 2nlog|\Sigma| + nlog(2\pi) \end{aligned}$$

Θ here are l_1, l_2, σ, and data here is $(X, \mathbf{y})$, $\mu = \mathbf{0}$

4. Given a start point of the parameters, we denote it as Θ_0, we aim to find the best parameters which minimize $L(\Theta, data)$. Below are the details of iterations when using SGD:
• choose a proper sample size n_1 and the step size $\alpha_\mathbf{k} \in \mathcal{R}^3$. The role of α_k here is to decide how long the parameters should go towards a certain direction and since in we have three parameters now, the length of α_k is three
Step 1: sample n_1 data from $(X, \mathbf{y})$, which forms a new subset $(X_{sub}, \mathbf{y}_{sub})$, where X_{sub} is a $n_1 \cdot p$ matrix and $\mathbf{y}_{sub} \in \mathcal{R}^{n_1}$

Step 2: Starting from $k = 0$, we calculate the gradient of $L(\Theta, data)$ when $\Theta = \Theta_0$ *and* $data = (X_{sub}, \mathbf{y}_{sub})$, i.e.

$$\triangledown L(\Theta = \Theta_0, data = (X_{sub}, \mathbf{y}_{sub}))$$

Step 3: Update Θ:

$$\Theta_{k+1} = \Theta_k - \alpha_k^T \nabla L(\Theta = \Theta_0, data = (X_{sub}, \mathbf{y}_{sub}))$$

Step 4: Compute the distance between Θ_{k+1} and Θ_k, i.e. $r = \|\Theta_{k+1} - \Theta_k\|_2$, if $r \geq threshold$, then we return to the first step and continue sampling. Otherwise, we stop and consider Θ_{k+1} as the best parameter. Here the threshold is usually 10^{-4} and sometimes would change on different data set. We denote the optimized parameters we get is Θ^*.

1.2 making prediction

Assume we have a training data set $(X, \mathbf{y})$ and testing data set $(X^*, \mathbf{y}^*)$

1. Based on the data set $(X, \mathbf{y})$, we could find the best parameters using the algorithm in **section 1.1**, we also denote it as $\Theta^* = (l_1^*, l_2^*, \sigma^*)$
2. Compute the $Kernel$ using Σ^*, which is

$$K(\mathbf{x_p}, \mathbf{x_q}) = {l_1^*}^2 \exp\left(-|\mathbf{x_p} - \mathbf{x_q}|^2 / 2{l_2^*}^2\right)$$

3. Now we can write the joint distribution of $\mathbf{y}$ and $\mathbf{y}^*$ as

$$\left[\begin{array}{c} \mathbf{y} \\ \mathbf{y}^* \end{array}\right] \sim \mathcal{N}\left(\mathbf{0}, \left[\begin{array}{cc} K(X,X) + {\sigma^*}^2 I & K\left(X, X^*\right) \\ K\left(X^*, X\right) & K\left(X^*, X^*\right) \end{array}\right]\right)$$

4. Deriving the conditional distribution, we arrive at the key predictive equations: $\mathbf{y}_* | X, \mathbf{y}, X_* \sim \mathcal{N}\left(\overline{\mathbf{y}}_*, \operatorname{cov}\left(\mathbf{y}_*\right)\right)$ where

$$\overline{\mathbf{y}}^* \triangleq \mathbb{E}\left[\mathbf{y}^* | X, \mathbf{y}, X^*\right] = K\left(X^*, X\right)\left[K(X,X) + {\sigma^*}^2 I\right]^{-1} \mathbf{y}$$

$$\operatorname{cov}\left(\mathbf{y}^*\right) = K\left(X^*, X^*\right) - K\left(X^*, X\right)\left[K(X,X) + {\sigma^*}^2 I\right]^{-1} K\left(X, X^*\right)$$

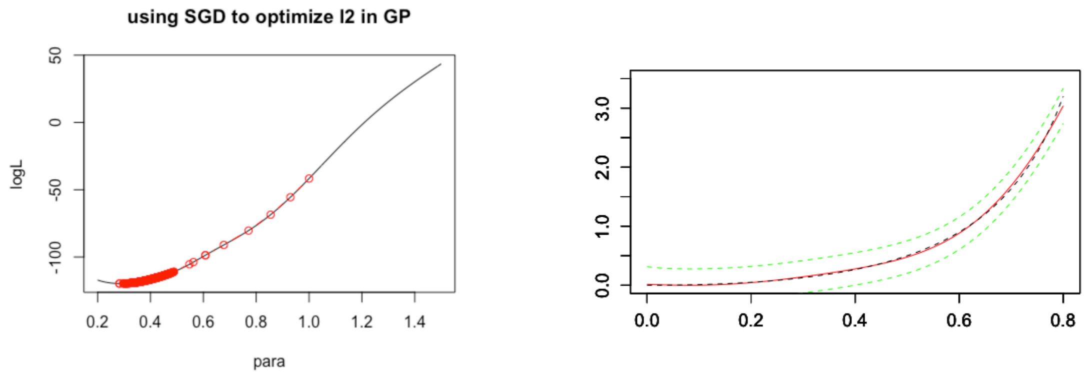
2 Eigenvalues and their Decay rates

For SE kernel in 1.1, the m-th eigenvalue is $\lambda_m = v\sqrt{2a/A}B^{m-1}$, where $a = 1/(4\sigma_\epsilon^2)$, $b = 1/(2\ell^2)$, $c = \sqrt{a^2 + 2ab}$, $A = a + b + c$ and $B = b/A$, ℓ is the length parameter, v is signal variance and σ_ϵ is the noise parameter. We can obtain $\sum_{m=M+1}^{\infty} \lambda_m = \frac{v\sqrt{2a}}{(1-B)\sqrt{A}} B^M$

For the Matérn $k + \frac{1}{2}$, $\lambda_m \asymp \frac{1}{m^{2k+2}}$. We can obtain $\sum_{m=M+1}^{\infty} \lambda_m = \mathcal{O}(\frac{1}{M^{2k+1}})$.

3 Simulation

- Simple case: one dimension, one parameter
  kernal $= 3^2 \cdot \exp\left(-0.5d^2/l^2\right) + I_2 \cdot 0.1^2$
  $y = x^2/(1-x)$
  learning - rate $= 0.003$, sample size $= 10$



(a) convergence of loglikelihood versus the value of l_2

(b) Prediction

Figure 1: From (a), we could see the direction and how long l_2 moved in each iteration. In (b), the black line is the true value of response variable in testing set y_{true}, and red one is the predicted value y_{pred}. Two green line are $y_{pred} \pm 3\sqrt{\widehat{var}(y)}$

- One dimension, three parameters:

we generate data from $\mathcal{N}(\mathbf{0}, \Sigma)$, where $\Sigma = 4^2 \cdot e^{\left(-D^2/2 \cdot 2^2\right)} + \text{diag}(0.1^2, n)$.Here D denote the distance between data points
number of training set: $n = 100$; number of testing set: $n_1 = 999$
Sample$-$size $= 4$; learning-rate: $\alpha_k = 0.08/ceiling(k/12)$
we aim to predict the parameters in $\Sigma = l_1^2 \cdot e^{\left(-D^2/2 \cdot l_2^2\right)} + \text{diag}(\sigma^2, n)$

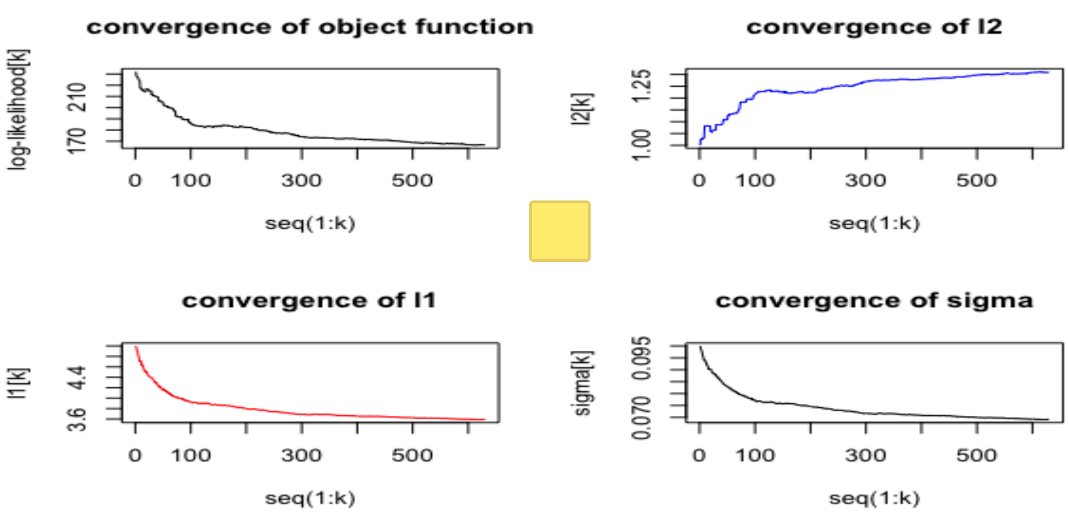


Figure 2: the convergence of three parameters

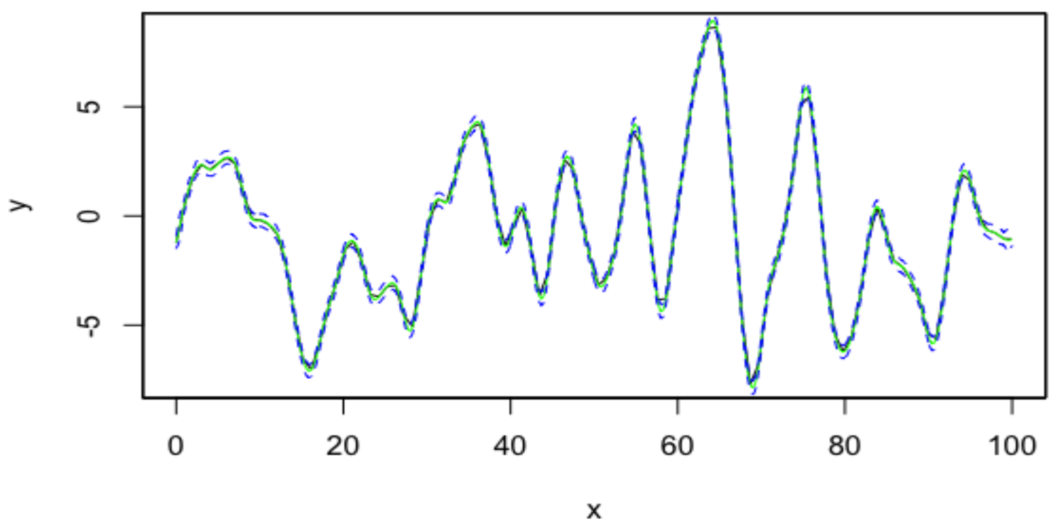


Figure 3: Making Prediction

• Three dimension, two parameters:
we generate data from $\mathcal{N}(\mathbf{0}, \Sigma)$, where $\Sigma = 4^2 \cdot e^{\left(-D^2/2\cdot 2^2\right)} + \text{diag}(0.001, n)$
$n = 20$, $Sample - size = 5$, $\alpha_k = 0.01/ceiling(k/20)$
we aim to predict the parameters in $\Sigma = l_1^2 \cdot e^{\left(-D^2/2\cdot l_2^2\right)} + \text{diag}(0.001, n)$

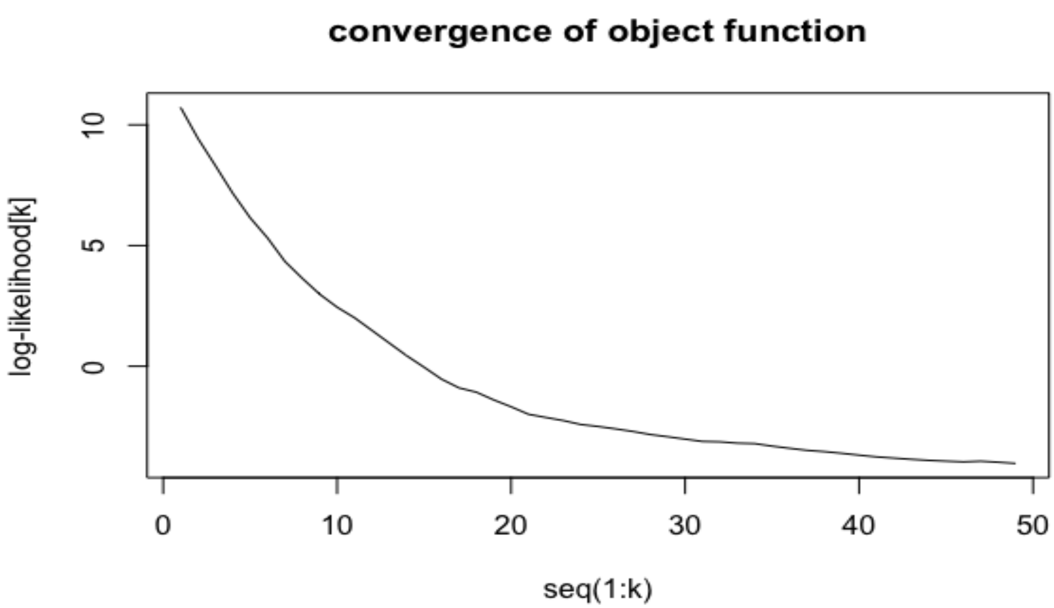
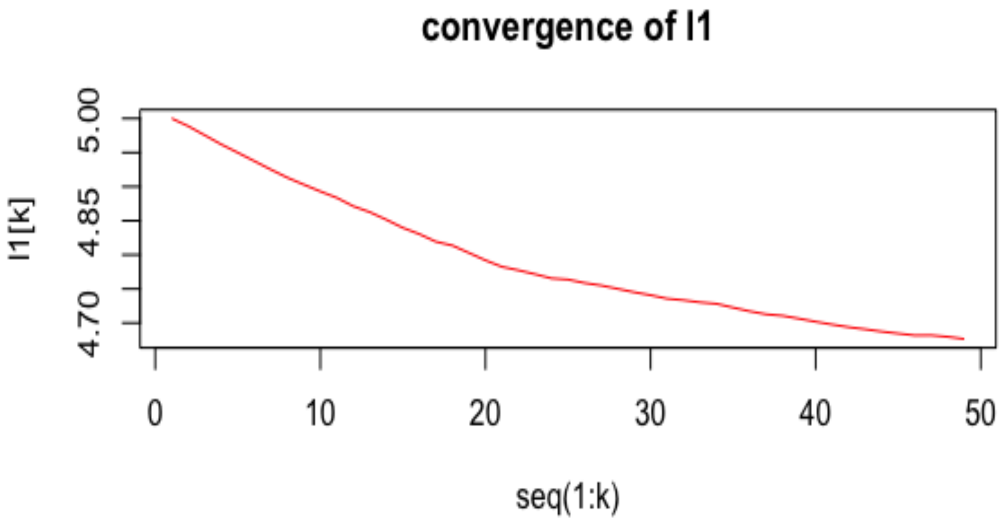


Figure 4: Convergence of log-likelihood versus step k



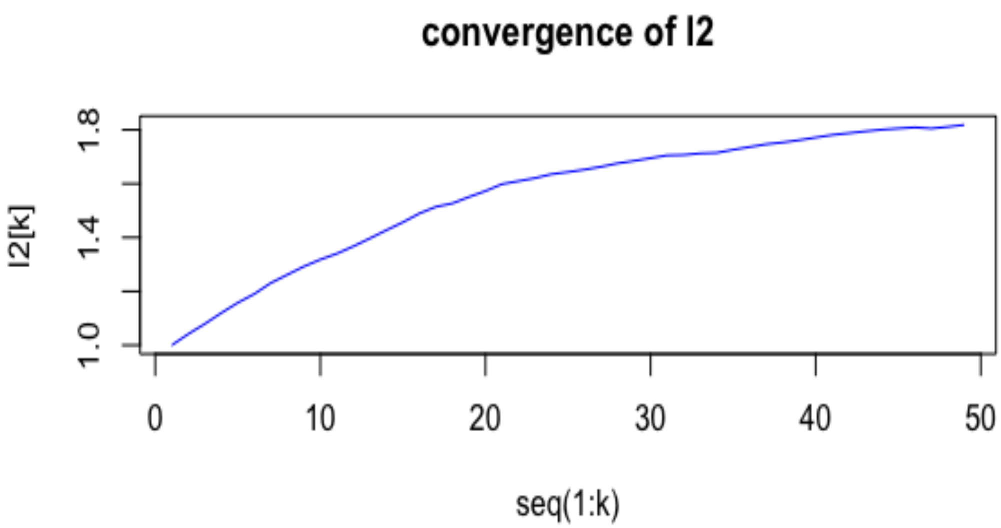
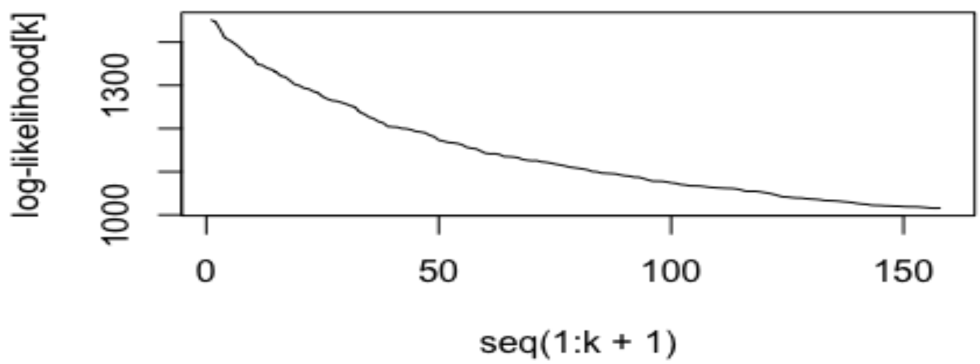


Figure 5: Convergence of parameters versus step k

• use Matern kernel $k_{\nu=3/2}(r) = \left(1+\frac{\sqrt{3}r}{4}\right)\exp\left(-\frac{\sqrt{3}r}{4}\right)$
$n = 100$, $p = 3$, $learning - rate = 0.01/ceiling(k/20)$
we aim to predict the parameters in $k_{\nu=3/2}(r) = \left(1+\frac{\sqrt{3}r}{\ell}\right)\exp\left(-\frac{\sqrt{3}r}{\ell}\right)$



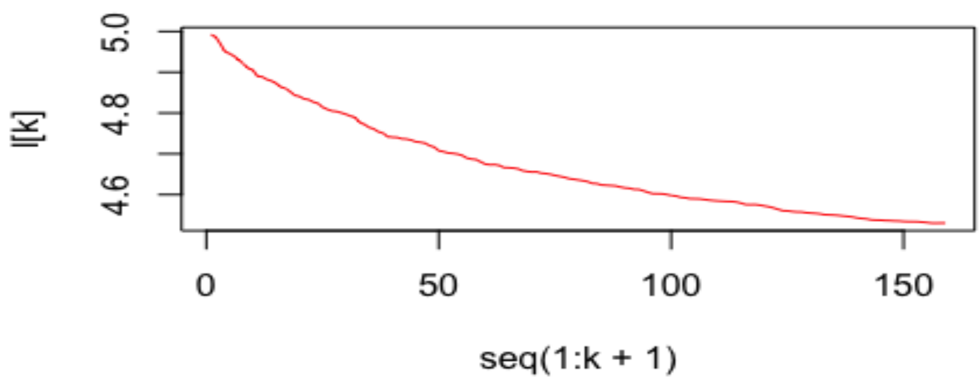


Figure 6: the convergence of loglikelihood and parameter using Matern Kernel

• Five dimension, six parameters:

We generate y from $\mathcal{N}(\mathbf{0}, \Sigma)$, and Σ is formed from

$$K(\mathbf{x}, \mathbf{x}') = 4^2 \cdot e^{-\frac{1}{2}\sum_{i=1}^{p}|x_i - x_i'|/l_i^2}$$

Here x_i denote the i_{th} element of $\mathbf{x}$ and x_i' denote the i_{th} element of $\mathbf{x}'$, assuming $l_1 = 1, l_2 = 2, l_3 = 3, l_4 = 4, l_5 = 2$ and $\sigma = 0.001$

$n = 200$, $Sample - size = 5$, $\alpha_k = 0.01/ceiling(k/20)$

we aim to predict the parameters in $\Sigma = 4^2 \cdot e^{-\frac{1}{2}\sum_{i=1}^{p}|x_i - x_i'|/l_i^2} + diag(\sigma^2, n)$, i.e. l_1, l_2, l_3, l_4, l_5 and σ

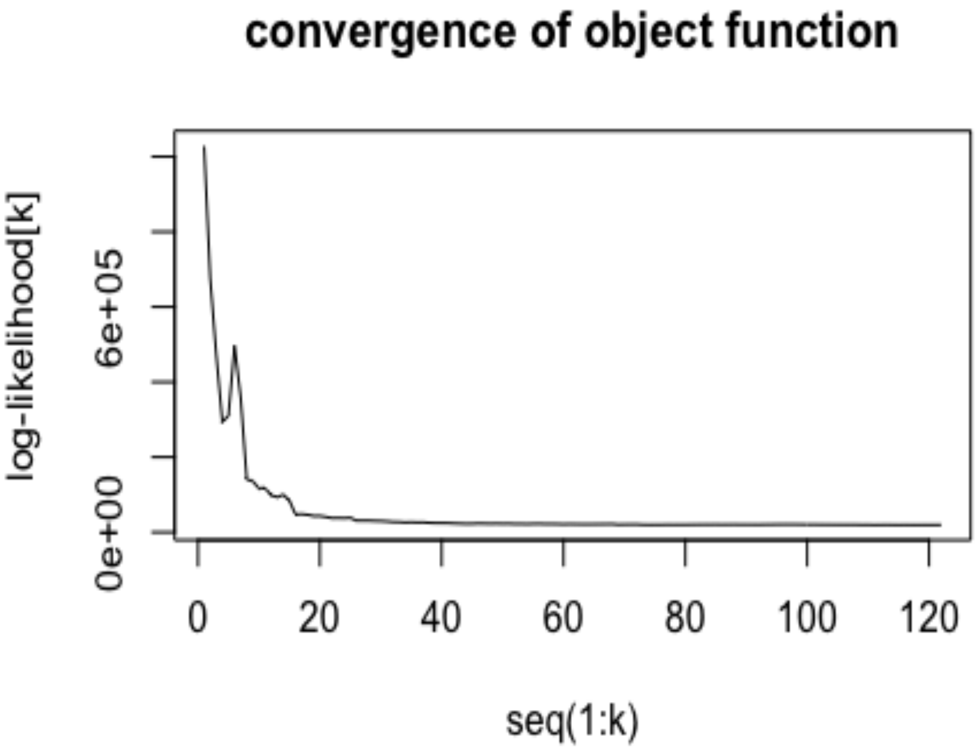


Figure 7: convergence of log-likelihood versus step k

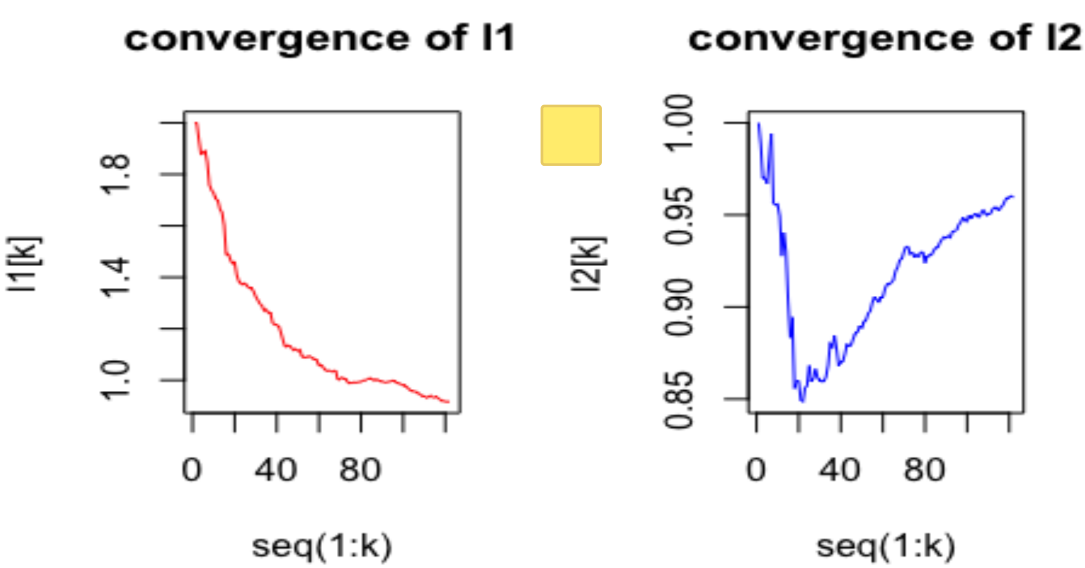


Figure 8: convergence of parameters l_1 and l_2 versus step

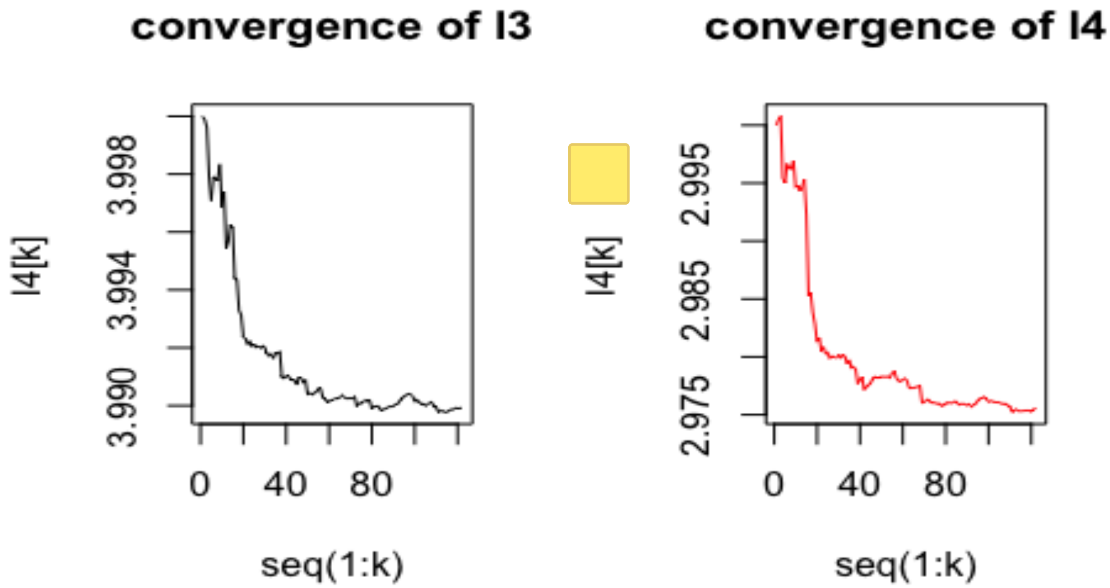


Figure 9: convergence of parameters l_3 and l_4 versus step

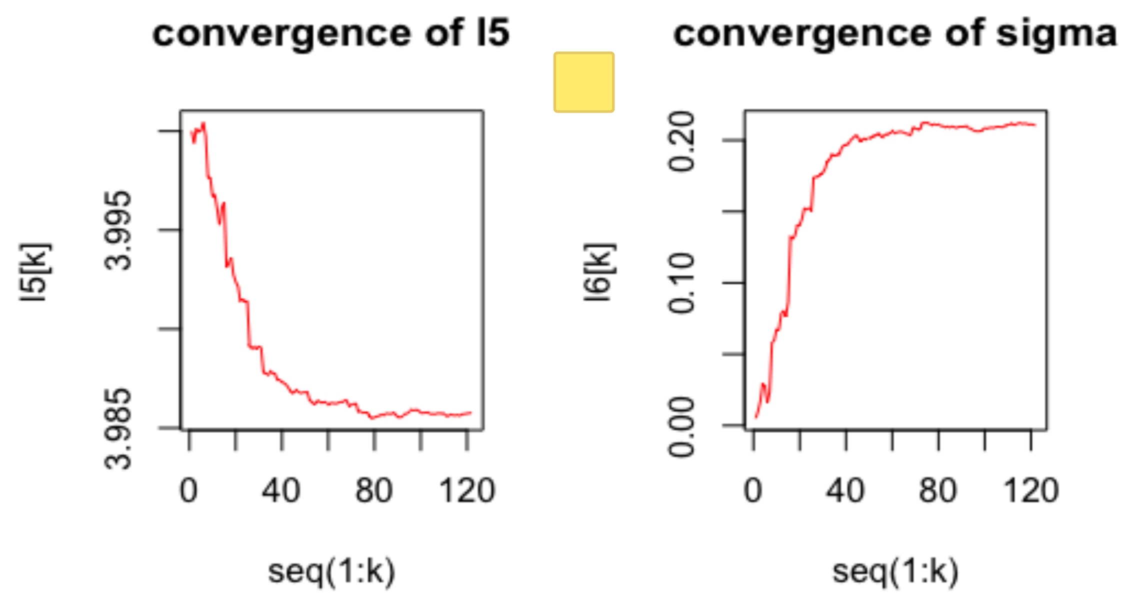


Figure 10: convergence of parameters l_5 and σ versus step