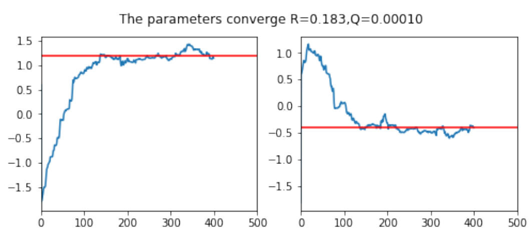
Kalman and Particle Filters; Online PCA Report

YANG LI, 31319653, yl13y19@soton.ac.uk

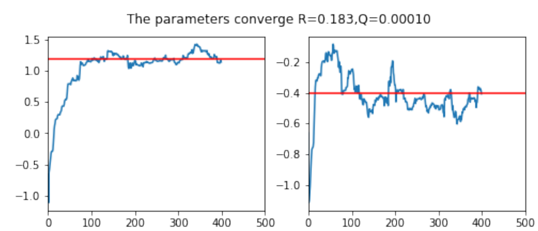
April 22, 2020

1 Tasks 1

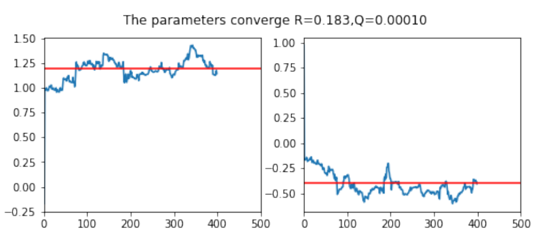
The parameters converge for five different initial conditions show in Fig. 1.



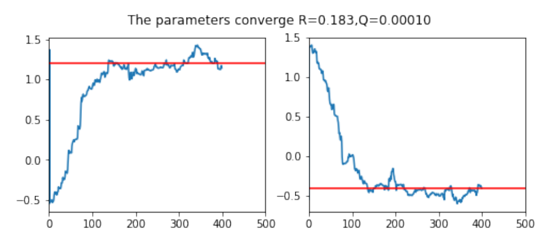
(a)



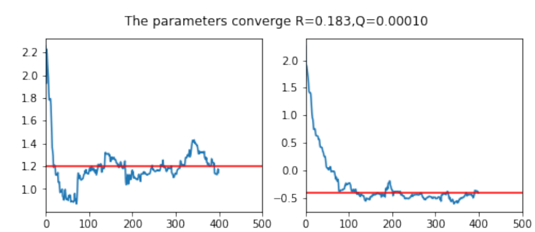
(b)



(c)



(d)

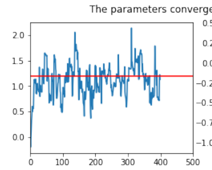


(e)

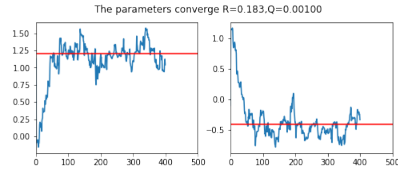
Figure 1: The parameters converge for five different initial conditions

The process noise bigger the vibration is stronger but the convergence speed is high, which shows in Fig. 2.

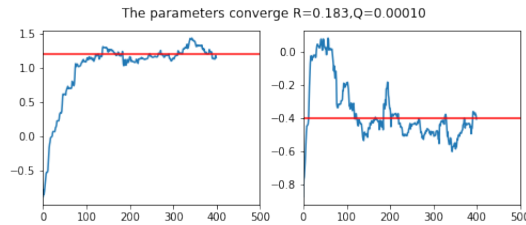
The process noise smaller the vibration is stronger but the convergence speed is high, which shows in Fig. 3.



(a)



(b)



(c)

Figure 2: The various values for the process noise

2 Tasks 2

2.1 Construct a dataset

The SIS suffers from degeneracy of particle weights shows in Fig. 4 (select some pictures).

I use resampling after every epoch of SIS, and the weights changes (the weighs are same after resampling, I pick the weights pictures are after once SIS and before next resampling), which shows in Fig. 5.

The initial particles and final particles show in Fig. 6.

3 Task 3

The class boundary converges to the true solution shows in Fig. 7.

EKL Algorithm:

Generate two classes point and target with noise: $\{Data_n, Target_n\}_{n=1}^{N}$

Initialize the $\theta(n-1|n-1)$, $P(n-1|n-1)$, R and Q.

FOR $i = 0 : 2 * N$:

Random select a data and correspond target from Data and Target

$\theta(n|n-1) = \theta(n-1|n-1)$

$P(n|n-1) = P(n-1|n-1) + Q$

$y(n) = 1/(1 + exp(-\theta(n|n-1)@data))$

$error = CrossEntropy(y(n), target)$

$covariance = data^T @P(n-1|n-1)@data + R$

$gain = P(n|n-1)@data^T$

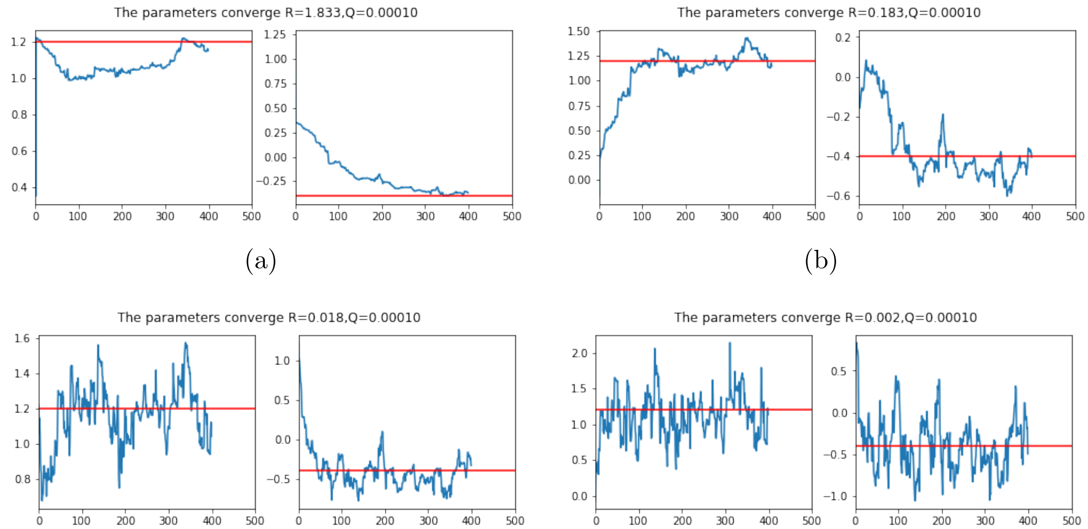
$\theta(n|n) = \theta(n|n-1) + gain * error$

$P(n|n) = (I - gain@data^T)@P(n|n-1)$

$\theta(n-1|n-1) = \theta(n|n)$

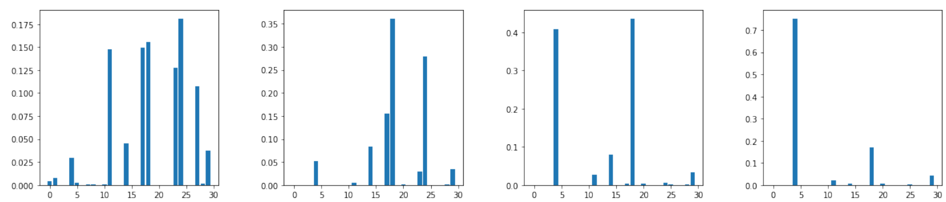
$P(n-1|n-1) = P(n|n)$

End For



(a) (b) (c) (d)

Figure 3: The various values for the measurement noise



(a) (b) (c) (d)

Figure 4: Degeneracy of particle weights

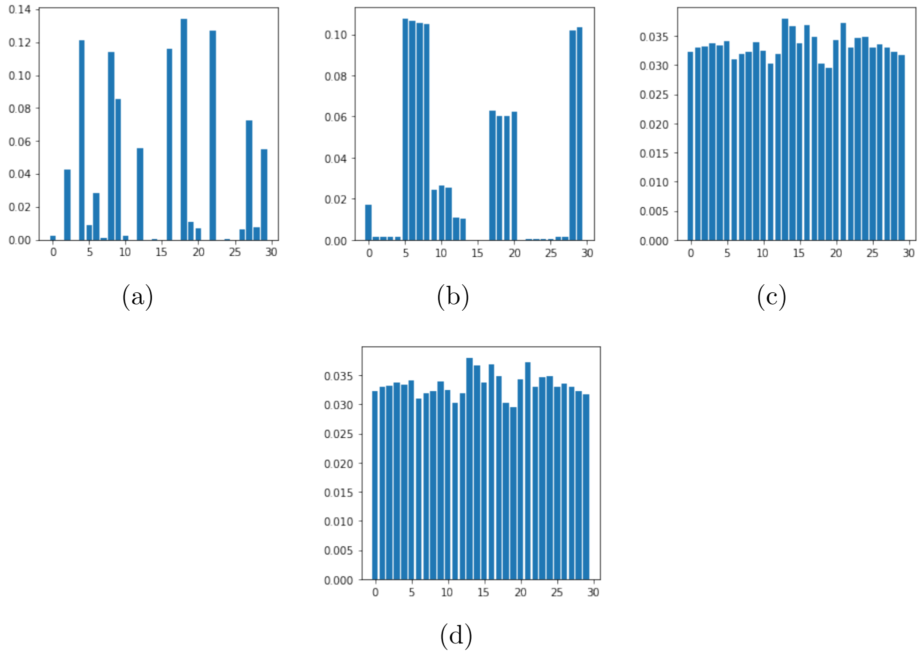
4 Task 4

Online PCA algorithm can be used for time-series and multivariate problems. For example, Beijing PM2.5 Data Set in UCI Repository. There are 13 attributes in the data set. Suppose I need to predict the future PM2.5, the data recording will not stop and the predict need to be visualised.

There are 13 attributes in the data set(No, year, month, day, hour, pm2.5, DEWP, TEMP, PRES, cbwd, lws, ls and lr). I might select pm2.5 as output (target) and other attributes as input except No. Each set of data as x_t, l as 8 and d as 11.

I think low dimensional representations might result the compressed information or the information which removed some noise of raw data.

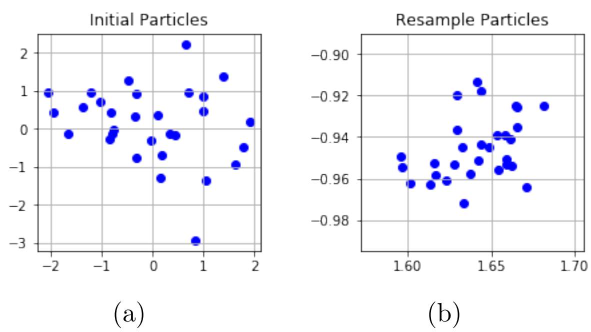
In these cases, online PCA not only reduces data double calculation but also reduce the computation(avoid curse of dimensionality). It use only $O(dl)$ floating point operations per input vector and uses only $O(dl)$ space. It avoids storing and manipulating the covariance matrix C.



(a) (b) (c)

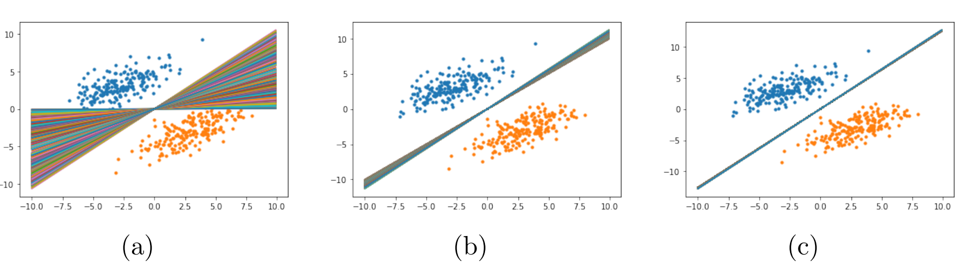
(d)

Figure 5: Weights of resampling



(a) (b)

Figure 6: Initial particles (a) and final particles (b)



(a) (b) (c)

Figure 7: The class boundary converges to the true solution for random initial situation