COMP6247 Reinforcement and Online Learning Lab 2 Report

MSc.Artificial Intelligence
Student: Wei Lou
wl4u19@soton.ac.uk
Supervisor: Dr Mahesan Niranjan

1 Introduction

THISlab is to understand and apply some online learning algorithm with supervised or unsupervised problems. Standard Kalman filter is optimal in estimating time-series state of linear models with Gaussian noise and extended Kalman filter is to deal with the nonlinear system. Particle filter works further to solve non-linear models with non-gaussian noise. When processing larger dataset, it's useful to reduce the dimension of the original data to fit the memory or computational constraint. Online PCA is one of the good methods to reduce noise and preprocess a big dataset to low dimension space in real-time.

2 Implementation

2.1 Task1: Kalman Filter

This task is to implement a Kalman filter to estimate the parameters of a synthetic second-order AR process. An autoregressive time series also called AR process of order p is given by a generating model:

$$s(n) = \sum_{k=1}^{p} a_k s(n-k) + v(n)$$

In this lab, p=2, N=400, using Gaussian random noise. The synthetic signal shows in Figure.1.

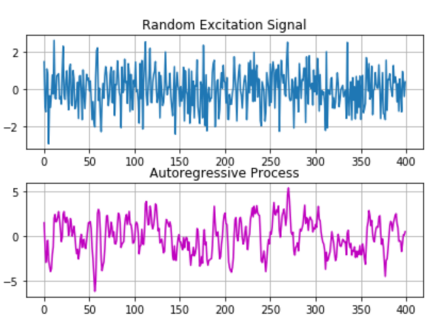

Fig. 1: Signal visualization

Then implement the Kalman filter, choosing different parameters R,Q would cause different converge curves.

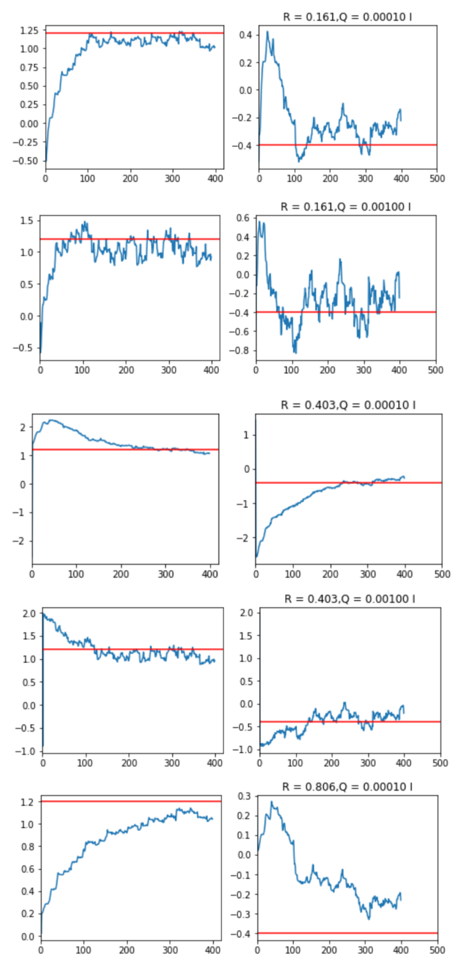

Fig. 2: Converge Curve of different R,Q

It's clear that if R becomes larger, the converging speed would slow down and an increase of Q would bring in more uncertainty, make the converge faster.

2.2 Task2. Particle Filter and Resampling

The particle filter is one of the methods of online bayesian inference. It uses sampling theory, sample particles from known distribution and calculates the particle weights

based on the likelihood between the known distribution and original data distribution. The process could be divided into 4 steps:
1.Initialize particles, in this lab, 30 particles of 2 dimensions are randomly picked.
2.Update the particles: add small random perturbation into the original particles.
3.Prediction and Likelihood: Prediction of next true value can be calculated by the last 2 real data x[0],x[1] and current particle. Then calculate the likelihood of estimation and real data.
4.Set particle weights: Particle weights are updated by multiply the likelihood and normalized by dividing by the sum of all weights.
This standard solution has a certain ability of estimation real data, however, there is a degeneracy problem for this method. After a few iterations, the algorithm would only focus those particles with very large weights and ignore all the other particles just like the following figure shows.

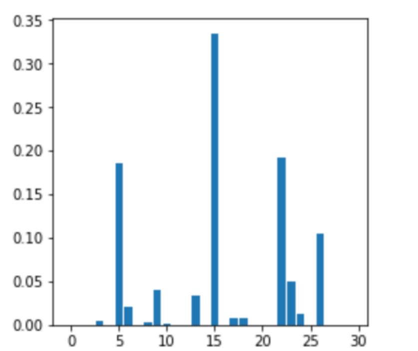

Fig. 3: Particle weight distribution

In order to solve this problem, a resampling algorithm can be used. The basic idea is to resample the particles whenever a significant degeneracy is observed. The particles with larger weights can be sampled many times. The weights show low variation in every iteration based on this algorithm.

2.3 Task3.Extended Kalman Filter

Because of the estimation usage of Kalman filter, it could perform logistic regression. The data used in this task is a synthetic data of 2-dimensions and 2-classes with distinct means and common covariance. Extended Kalman filter works for estimating non-linear function. The basic implement is similar to Kalman filter, except the yh now becomes:

$$\frac{1}{1+\exp\left(-\boldsymbol{\theta}^T\boldsymbol{x}\right)}$$

The converge of this algorithm is as follows: m1 = [-5,5], m2= [5,-5], C=[[2,1],[1,2]]

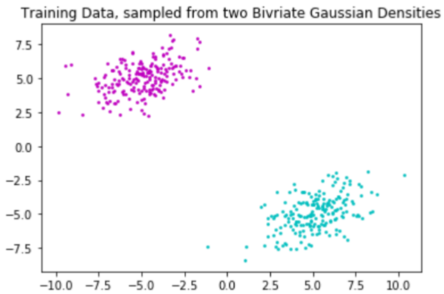

Fig. 4: Training Data

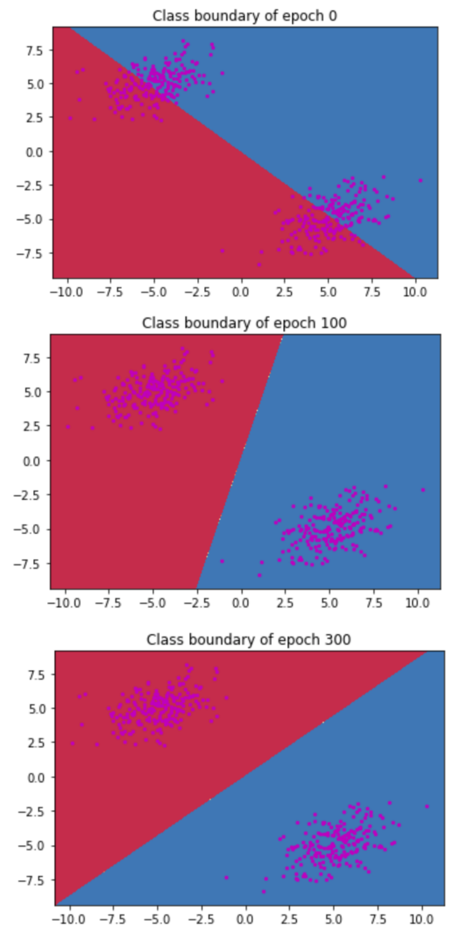

Fig. 5: Converge process of different epoch

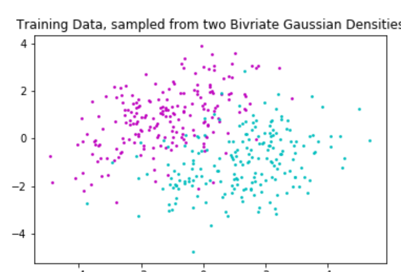

Fig. 6: Training Data

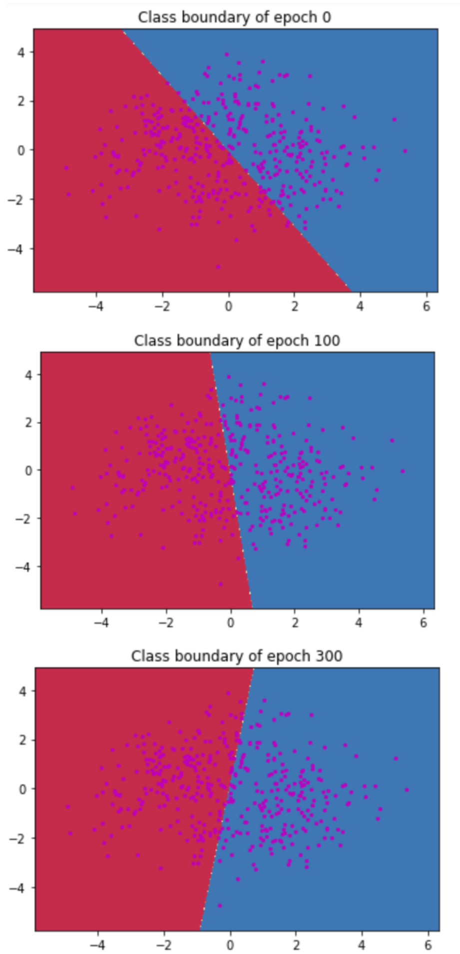

Fig. 7: Converge process of different epoch

2.3.1 Task4: Online PCA algorithm

Online PCA algorithm can decrease the data dimension in order to reduce computational cost and energy cost. Taken Kaggle Chest X-Ray Images as an example, the original dimension is 5863*1850*1300 because the image size is 1850*1300, the image feature is very large even after feature extraction. The data size could be reduced by using the PCA method. Also using online PCA cost much less than an offline PCA which is more useful for a real-time processing system. The parameters need to be specified are m: the original dimension, K: the dimension of after PCA, U—the principal components matrix in every iteration, error—the error of applying current U to each new data, U'—new principal components matrix whose dimension is m*(K+1). The low dimensional representation of the result should be data numbers * new reduced dimension number K. And the computational saving should be (m*m-(K+1)*(K+1)) because m»K, there is a huge saving by using Online PCA.