Probabilistic Models (HMM) and Change Point Detection in Time Series Data

Li Hsin Cheng, Linxiao Bai, Zhongda Su
University of Rochester
Rochester, NY

Abstract – This report briefly explains a methodology of using MAP decision and EM algorithm to detect the probabilistic model change point of a time series data. Different probabilistic models are tested in simulated dataset and result shows that although in general the accuracy depends on initial conditions as well as the similarity of the real model, this method gives promising result under different time setting.

Keywords – HMM, change point detection; probabilistic model; time series analysis, MAP estimator

I. INTRODUCTION

Time series data and its associated random process are particularly interesting in the setting of real life. Real life data is generated at the fashion of time increasing order. In fact, studies of determination of the particular model that results the observation has been a hard question. Common approaches include using MAP decision to estimate the best parameter set for a given model, minimize cost function based model studies, and other splining method based on cost function such as cumulative sum and test statistics. The realization of time series is a dynamic random process determines the nature that at different time point model may change accordingly. This nature explains the poor performance of single model based approaches on real life data.

It is known that given a correct guess of the actual model, a MAP decision will give the best estimator of parameters. It would seem easy to simply try out different splitting point and fit separately two guesses of model and select the result that yields the highest likelihood. However, this exhausting method could run into computational problem as time space and observation dimension increases. Given such context, an improvement could be using EM algorithm to obtain the local best splitting decision given an initial splitting guess. Surprisingly, without traversing the data too many times, the algorithm generates good guesses of the actual splitting point of the data. [2][3][4][5][6]

II. MOTIVATION

HMM is well known for describing the pattern of the stationary distribution; however, due to the rapid-changing pattern and the range of the sequence data, it often requires large state space to fully characterize the pattern of the sequential data. When the state space grows, the probability of some states become extremely small and thus becomes more and more difficult to converge, or worse, yield wrong result. The methodology presented illustrate a way of separating the sequence data into multiple HMMs. Each HMM can characterize a period of sequence data and thus need far less states and gives equally accurate result.

III. METHODOLOGY

- ***General Introduction***

We start off with an initial guess of the general form of the model, so the posterior probability can be computed. As well as a guess of initial point. For the sake of of convenience and illustration, one dimension of Gaussian was firstly chosen as implementation as the sufficient statistics of model parameters are easily computed. Then a more complex probabilistic model HMM are implemented to monitor the behavior. Finally, we introduce a method based on statistics test and threshold to compare performance.

- ***Mathematics Explanation.***

Starting off with an initial guess of the splitting point ***t***, parameters of the left hand side model is determined by MAP decision:

$$\boldsymbol{\theta_1} = \arg\max_{\boldsymbol{\theta 1}} P(\boldsymbol{\theta_1} \mid X_1{:}Xt)$$

Similarly, for the right hand side model:

$$\boldsymbol{\theta_2} = \arg\max_{\boldsymbol{\theta 2}} P(\boldsymbol{\theta_2} \mid Xt{:}Xn)$$

Using $Model_1$ and $Model_2$ just discovered, another MAP/ML (since evaluation function are the same) decision can be made about the new ***t*** :

$$\boldsymbol{t'} = \arg\max_{t'} P(X_1{:}Xt \mid \boldsymbol{\theta_1}) \times P(Xt{:}Xn \mid \boldsymbol{\theta_2})$$

The algorithm iterates until convergence on t or reaches max iteration.

This is a typical EM problem and by as long as E step and M step are using MAP decisions, a convergence will be guaranteed. Speed of the convergence is not clear to us, and will not be considered.

- ***Data Simulation.***

Data is simulated based on which ever probabilistic model of choice with random parameter, and random splitting point hidden from the algorithm. At testing step in order the simplify cases and control variable, splitting point of test sets is fixed to be a certain point.

- ***Test Result.***

Based on simulated data of different random parameters, predicted splitting point is compared with the actual value. Further analysis of performance will be analyzed. For the sake of simplification, try-out of initial points or other initial values may or may not be included.

IV. ANALYSIS

A. Gaussian Model.

Gaussian model with splitting point is a great starting point for the following reasons:

- Sufficient statistics of Gaussian parameter exist and can be computed at linear time:

  $mu = E(x_1{:}x_t)$

  $sigma = \sqrt{var(x1{:}xt)}$

- Using dynamic programming, MAP of ***t*** can be computed at constant time.

- Gaussian distribution is a common model that can be broadly applied. Especially in the application of study of white noise. Common cases will be analyzed.

The algorithm is implemented in Python. Exactly 100 test data of length of 1000 are generated. Each data has a set of randomly chosen parameter μ_1, μ_2, σ_1, σ_2. The division of the two models are at exactly at 200 for all data. This setting is to control irrelevant variables.

Error at a particular trial is defined as:

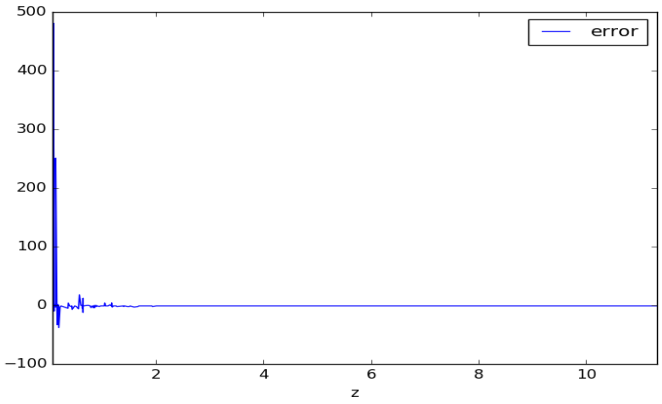
$$\boldsymbol{Error = t_{predicted} - t_{real}}$$

To further analysis the impact of different combination of real models on the performance of the algorithm, the "Mushiness" of a pair of Gaussian distribution is defined as:

$$\boldsymbol{Z} = \frac{\mu 1 - \mu 2}{\sqrt{\sigma 1^2 + \sigma 2^2}}$$

This definition is inspired by the form of test statistics of two sample t test.

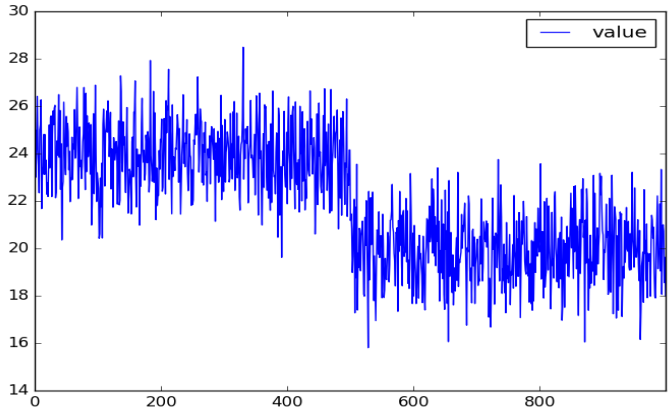
The simulation result is plotted as below:



The result shows that as the mushiness decreases, the error quickly converge to 0, yielding a great result. That is to say as the two model before and after splitting become easily spreadable, the EM splitting point detection algorithm almost surely gives the truth prediction. However, if mushiness exceeds certain threshold, in this case $z<2$, the splitting result is not guaranteed.

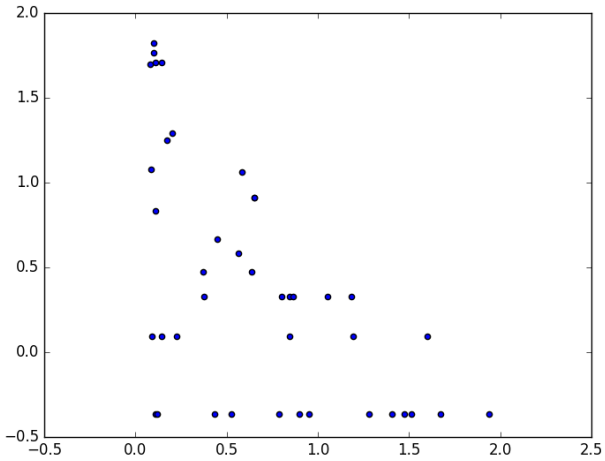
The threshold around $z>2$ is a guarantee of the global optima with no exception in 100 trails. This observation is critical. If it holds true for larger size trails, it indicates that the Mushiness index proposed is a sufficient deterministic index for global optima. Sine this observation should hold true for any Gaussian model, the generalization is very great.

To visualize spited signal of a pair of Gaussian with probability of z=2, arbitrary parameters μ_1=24, μ_2=20, σ_1,=σ_2=$\sqrt{2}$ is generated:



Keep in mind that other parameter might interfere with the global optima promise such as the time signal exists. This interference will not be studied. In the simulated data, the length of each signal lasts at least 200 observations.

To further analyze the rate that error converges with z, log function is applied to error twice, and plotted against z:



The result shows that the convergence is at least super linear.

B. *Hidden Marcov Model(HMM).*

We can determine the parameters of an HMM by using maximum likelihood.

$$p(X|\theta) = \sum_{z} P(X, Z|\theta]$$

However, this simply way is hard to calculate since we don't know Z and parameters. We can use EM algorithm to find efficient framework to maximize the likelihood function mentioned above.

In the E step, we use the old parameter θ^{old} to find the latent variables $p(Z|X, \theta^{old})$, then we can calculate the complete-data log likelihood estimation. This step is called expectation step. We have the following equation:

$Q(\theta, \theta^{old}) = \sum_z p(Z|X, \theta^{old}) lnP(X, Z|\theta)$

In the M step, we can get new parameters by getting maximization of Q function.

$$\theta^{new} = \arg\max_{\theta} Q(\theta, \theta^{old})$$

For convenience, we introduce new symbols to represents complex equation.

$$\gamma(z_n) = p(Zn|X, \theta^{old})$$

$$\xi(z_{n-1}, z_n) = p(z_{n-1}, z_n|X, \theta^{old})$$

Forward and backward are introduced to solve the computation complexity problem.

We can have $\alpha(z_n) \equiv p(x_{1,\dots} x_n, z_n)$

which means the joint probability of observations and the value of zn.

$\beta(zn) \equiv p(xn + 1, \dots xN|zn)$, which represents the probability of the future observation given the current state zn.

Gamma equals the multiplication of alpha and beta divided by probability of observation X. Recursively, we can have

$$\alpha(z_n) = p(x_n|z_n) \sum_{z_{n-1}} \alpha(z_{n-1}) p(z_n|z_n - 1)$$

$$\beta(z_n) = \sum_{z_{n+1}} \beta(z_{n+1})\, p(x_n + 1|z_n + 1) p(z_n + 1|z_n)$$

$$p(x) = \sum_{Zn} \alpha(z_n)\beta(z_n)$$

By setting n=N, we can have

$$p(x) = \sum_{z_n} \alpha(z_N)$$

Now let's talk about ξ, which correspond to the conditional probability of $p(z_{n-1}, z_n|X) = \frac{p(X|z_{n-1}, z_n) p(z_{n-1}, z_n)}{p(X)}$ by using bayes' rule

$= \frac{p(x_1,\dots,x_n) p(x_{n+1},\dots,x_N|z_n) p(z_n|z_{n-1}) p(z_{n-1})}{p(X)}$ by conditioning on z_n

$= \frac{\alpha(z_{n-1}) p(x_n|z_n) p(z_n|z_{n-1}) \beta(z_n)}{p(X)}$.

To use EM algorithm to train HMM, we first choose initialization of parameters. The transition probability matrix and initial state distribution usually follow uniform distribution. The emission function will dependent on the observations' distribution. Then we run forward and backward recursion to get $\gamma(z_n)$ and $\xi(z_{n-1}, z_n)$, which also gives us the likelihood function. This completes the E step. Then we can use M-step to get new θ by using maximizing the Q function with respect to π, A and $\emptyset$ respectively.

One problem with this method is that these probabilities are often

significantly small, which can cause underflow. To address this problem we can use normalized version of α which equals to $\frac{\alpha(z_n)}{p(x_1,\dots,x_n)}$. We also define scaling factors by conditional distributions c_n=p $(x_n|x_1,\dots,x_{n-1})$. Then we can have p $(x_1,\dots,x_n)$= $\prod_{m=1}^{n} c_m$.

$\alpha(z_n) = p(z_n|x_1,\dots,x_n)p(x_1,\dots,x_n) = \prod_{m=1}^{n} c_m\, \hat{\alpha}(z_n)$.

Similarly, we can rescale β,γ,ξ.

C. HMM Change Point Analysiss

The initialization of EM algorithm is the most important part of the methodology. Since EM does not guarantee global optimum, it is likely to fall into local optimum if we choose the initial point randomly. To avoid the problem, we use the technique discussed in [1] as a guess of our initial point. The test statistics follow the repeated sequential probability ratio (SPRT) test. The idea is similar to the CUMSUM Page's test except that the cumulative function is the probability function of HMM. We define the likelihood function:

$$L(k) = L_1^k = \sum_{i=1}^{k} \left(\ln\left(\frac{f_K(x_i)}{f_H(x_i)}\right)\right) ;$$

$$N = \arg\min\{S_n > h\};$$

$$S_n = \max\{0, S_{n-1} + g(x_n)\};$$

$$g(x_n) = \ln\left(\frac{f_K(x_n)}{f_H(x_n)}\right);$$

where k is the time at which the transition starts, f_K is the first distribution, f_H is the second distribution, S_n is the cumulative likelihood $g(x_i)$ from i=1~n and h is the threshold at which the CUMSUM test meets its criteria.

In HMM, the test statistics of $\ln\left(\frac{f_K(x_n)}{f_H(x_n)}\right)$ can be defined as follows:

$$f(x_t|x_{t-1},\dots,x_1) = f(x_t|x_{t-1},\dots,x_1,\theta) = p(x_t|x_{t-1},\dots,x_1,\theta) = \frac{\sum_{i=1}^{S}\alpha_t(i)}{\sum_{i=1}^{S}\alpha_{t-1}(i)}$$

where α_t is the forward joint probability of the states and previous observed event till t and S is the state space.

Following the SPRT test, we can have the approximate initial guess of the splitting point given that we first assume the sequence data to have split in the middle. Figure1 is the cumulative test statistics of figure2(c), the threshold is set to 6. The result serves as the initial starting point for the EM algorithm to minimize the probability of converging to local minimum, which is very likely to happen in this methodology.

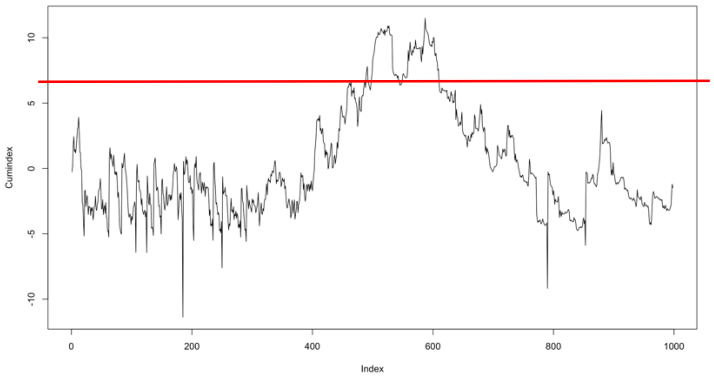


figure1

The EM methodology we developed is as follows:

x_i is the observed sequence, i=1~T, tp is the transition point of the sequence, θ_1 is the HMM1 model parameter, θ_2 is the HMM2 model parameter and α_t is the forward joint probability of the states and previous observed event till t and S is the state space.

Initialization: the initialization point is found in the previous step.

$$\mathrm{L}(\mathrm{tp}, \theta, x) = \mathrm{p}\left(x_{tp}, x_{t-1}, \dots, x_1 \middle| \theta_1\right) \mathrm{p}(x_t, x_{t-1}, \dots, x_1 | \theta_2)$$

$$= \left(\sum_{i=1}^{S} \alpha_{tp-1}(i) \right) \left(\sum_{i=1}^{S} \alpha_T(i) \right)$$

Maximize:

$$L^{n+1}(\mathrm{tp}_{n+1}, \theta, x) = argmax\ (L^n(\mathrm{tp}_n, \theta, x))$$

V. DATA SIMULATION AND TEST RESULT

We first generate simple Gaussian distributed time series of 2000 discrete points, and the transition point is 1000. That is to say, there is one Gaussian distributed series before 1000 and another Gaussian distributed series after 1000. The signal is as shown in figure 1(a).

Additionally, more complex distribution including mixture of Gaussians, Random distribution and Markovian distribution with states are generated in figure 1(b)(c)(d). The validation process is iterated using the EM algorithm as discussed above, and the initial point is selected between 800~1300. Figure 2 shows the likelihood function L(t, **θ**) with respect to time interval centered around 1000. The Likelihood function L(t, **θ**) is plotted as y-axis whereas x axis is the discrete time point centering around 1000. The overall time scale depiction is as shown in figure 3. Figure 4 (a)(b)(c)(d) is the calculated transition point and errors corresponding to figure1(a)(b)(c) and (d), respectively.

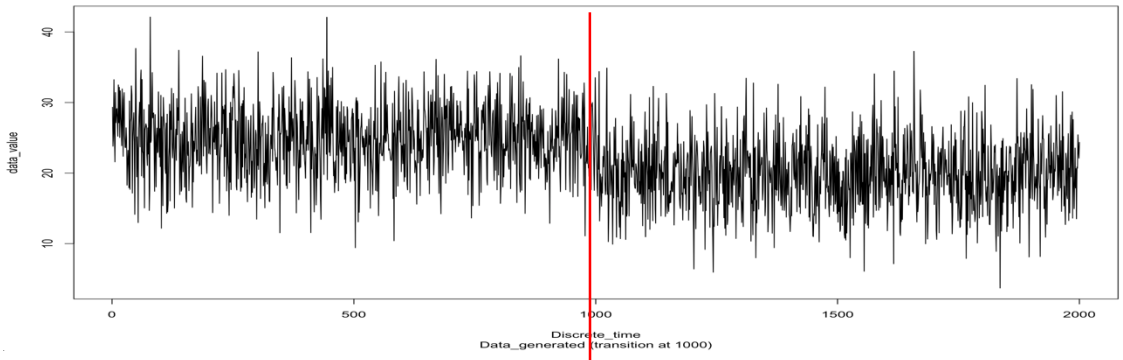


figure2(a)

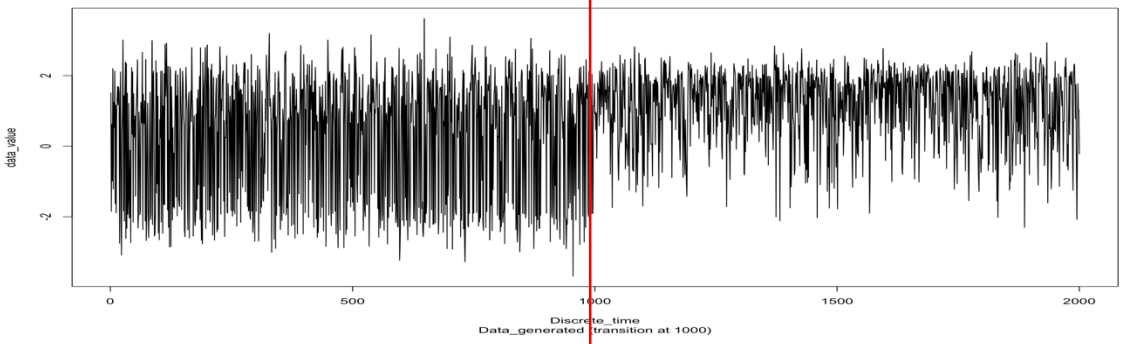


figure2(b)

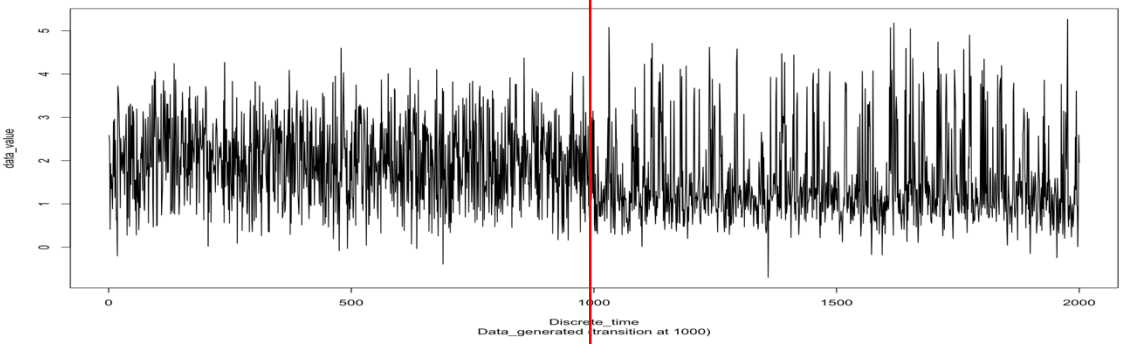


figure2(c)

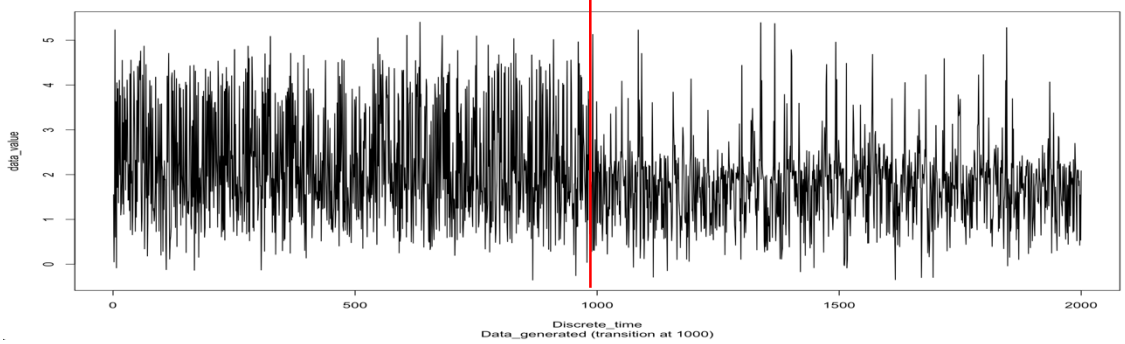


figure2(d)

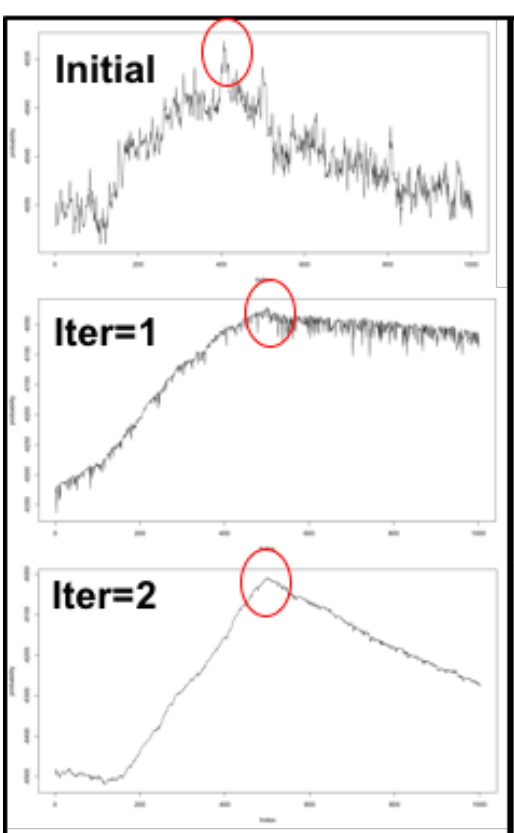


figure3(a)

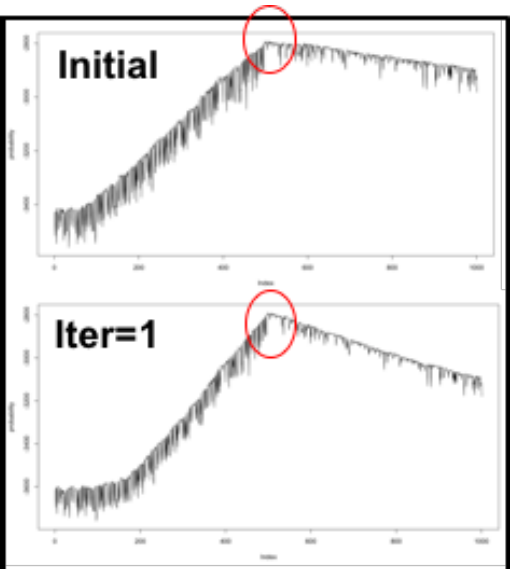
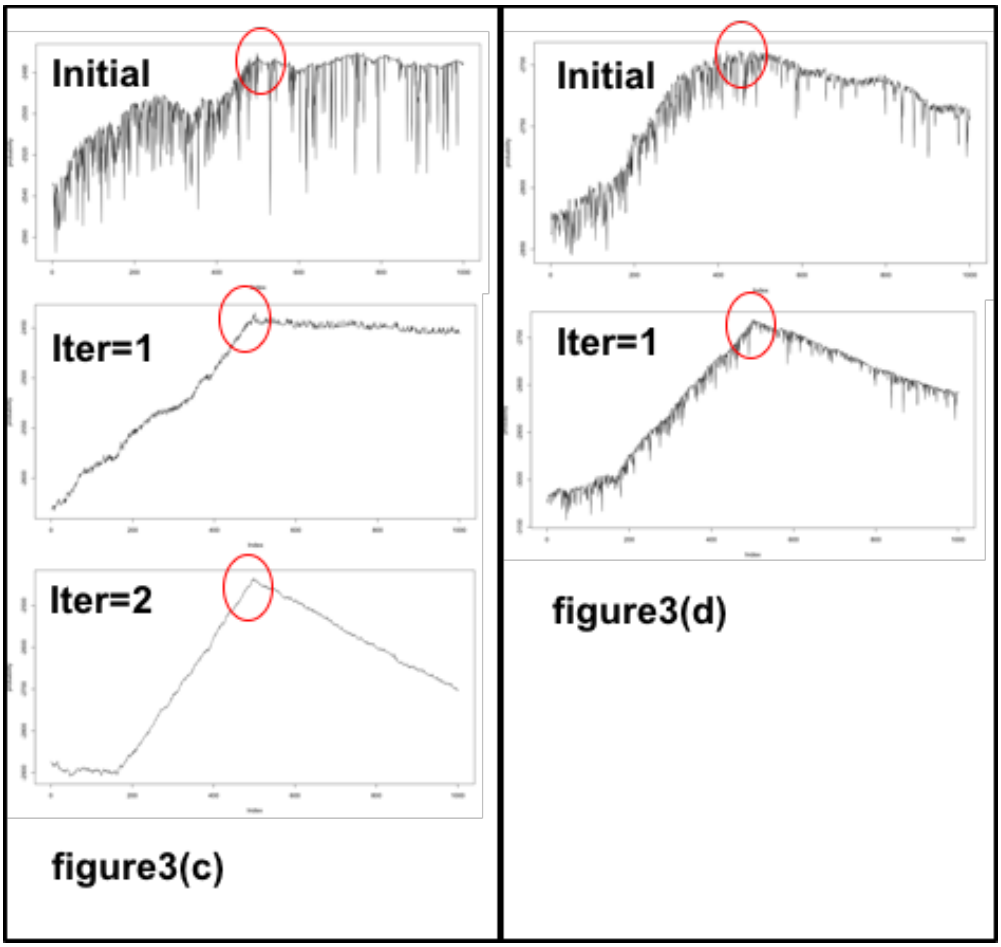


figure3(b)


Initial
Iter=1
Iter=2
figure3(c)
Initial
Iter=1
figure3(d)

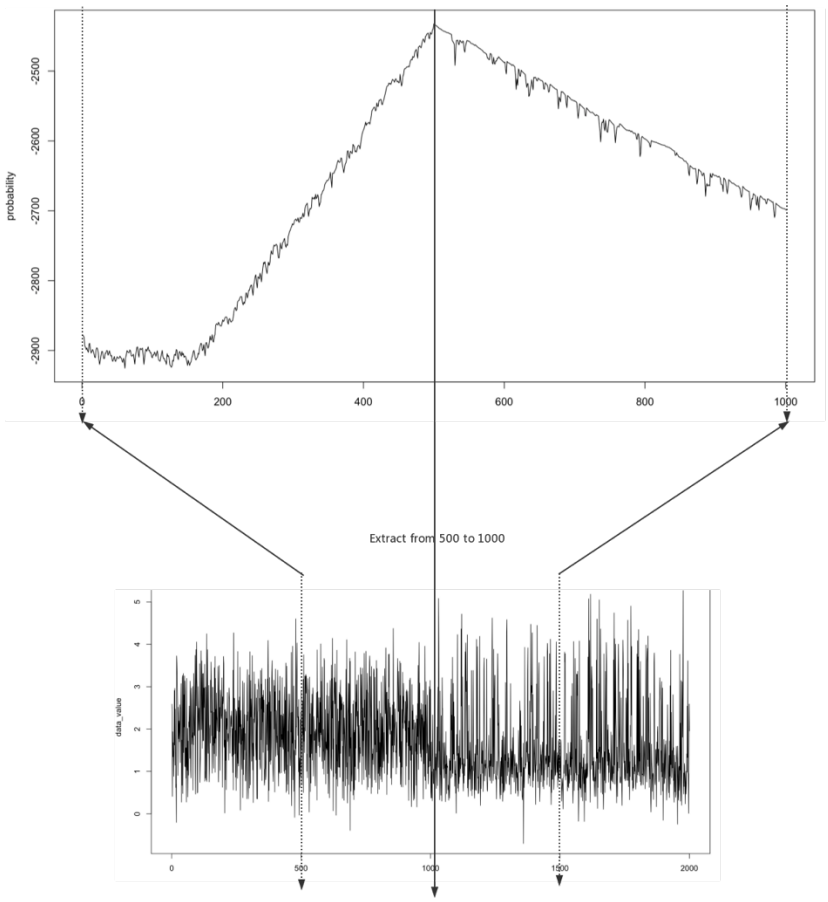


figure6

Initial_Point	Transition_Point (Discrete_Time_Index)	Error
900	1001	1
1000	1001	1
1100	1002	2

figure5(a)

Initial_Point	Transition_Point (Discrete_Time_Index)	Error
900	999	-1
1000	999	-1
1100	999	-1

figure5(b)

Initial_Point	Transition_Point (Discrete_Time_Index)	Error
800	1000	0
900	1000	0
1000	1001	1
1100	1001	1
1200	1001	1
1300	1001	1

figure5(c)

Initial_Point	Transition_Point (Discrete_Time_Index)	Error
800	1003	3
900	1001	1
1000	1001	1
1100	1003	3
1200	1001	1
1300	1003	3

figure5(d)

IV. Conclusion

The EM change point detection algorithm shows a great result on the simulated data. In the case of Gaussian model, if certain criterion of the model is satisfied ("mushiness">2), of detection error converges to 0 regardless of the initial point. Also, a shallow analysis of relationship between "mushiness" and error is analyzed. The result shows that predict error of only one initializing point converge to zero super linearly of mushiness. For more general case, HMM is analyzed with maximum likelihood of both model parameter of observed data. It shows promising result although the initial point is the main factor for converging to local optimum.

V. Future Work:

Since it is easy to have the function fall into local optimum given wrong initial point, it is important to set the good initial point. Although we have use PRTS to get the rough initial point, the result sometime still may fall into local optimum and yields wrong result. One possible way is to randomly generated starting points, and take the highest likelihood. There is also severity of local maxima for hierarchical latent class (HLC) models [7] to be used in future wosrk.

VI. References:

1. *B. Chen and P. Willett, "Superimposed HMM transient detection via target tracking ideas," IEEE Trans. Aerospace and Electronic Systems, vol. 37, no. 3, pp. 946-956, September 2001.*
2. *B. Chen and P. Willett, "Detection of hidden Markov model transient signals,"IEEE Trans. Aerospace and Electronic Systems, vol. 36, no. 4, pp. 1253-1268, Oct, 2000.*
3. *Christopher Bishop Pattern Recognition and Machine Learning, chapter 13, pp.605-644.*
4. *Ingmar Visser, University of Amsterdam, Seven things to remember about hidden Markov models: A tutorial on Markovian models for time series, Journal of Mathematical Psychology · December 2011.*
5. *Jan Bulla, Ingo Bulla , Oleg Nenadić , hsmm — An R package for analyzing hidden semi-Markov models, Computational Statistics and Data Analysis, 2008, August*
6. *Chen, J., & Gupta, A. K. (2000). Parametric statistical change point analysis. Boston:*

Birkhäuser.

7. Yi Wang and Nevin L. Zhang , "In Proceedings of the 3rd European Workshop on Probabilistic Graphical Models (PGM), 301-308."

(2) R packages

1. *Jared O'Connell, Inference for Hidden Markov and Semi-Markov Models, Package 'mhsmm'. https://cran.r-project.org/web/packages/mhsmm/index.html*

2. *Ingmar Visser, Dependent Mixture Models - Hidden Markov Models of GLMs and Other Distributions in S4, http://depmix.r-forge.r-project.org/*

3. *David Harte, Hidden Markov Models, Package 'HiddenMarkov', https://cran.r-project.org/web/packages/HiddenMarkov/HiddenMarkov.pdf*