Homework 4

Vladimir Frants, Yunhua Zhao, Mohamed Ben Zid

October 2020

Exercise 5.1

a)

Argue by using results of Chapter 1, that the optimal value θ^* the constraint must be active, that is, $L(\theta^*) = \alpha$.

Consecutive service times of each server are denoted by $\{S_n(\theta)\}$ and are independent with a general distribution with $\mathbb{E}[S_n(\theta)] = \theta$. The problem is to minimize the cost $C(\theta) = \frac{1}{\theta^2}$ which is a decreasing function of θ subject to the constraint $L(\theta) = \mathbb{P}(W(\theta) > w) \leq \alpha$, where $W(\theta)$ is a random variable with the stationary waiting time distribution and $\alpha \in (0, 1)$.

Let's assume that the inter-arrival times $\{A_n\}$, satisfy $[A_i] < \infty$, $Var(A_i) < \infty$, and that the service time distribution $W(\theta)$ has a Lebesgue density which is continuously differentiable in θ.

Then consecutive waiting times of customers constitute a Markov process given by the recursion:

$$W_n(\theta) = (W_{n-1}(\theta) + S_{n-1}(\theta) - A_n)_+$$

where $(x)_+ = max(0, x)$ is the positive part of the number x, W_n is the waiting time of customer n, and $W_0(\theta) = S_0(\theta) = 0$ which gives $W_1(\theta) = 0$ as it should because the first customer arriving to an empty system experiences no waiting.

The function $C(\theta) = \frac{1}{\theta^2}$ is decreasing, so we need to maximize θ satisfying the constraint. The probability to wait more than w is an increasing function of θ, so exists $0 < \theta^* < 1$ such that $L(\theta^*) = \alpha$. It means for $\theta \geq \theta^*, L(\theta) > \alpha$ the constraint is unfeasible. So, the solution for the constrained problem is θ^*.

b)

Due to *(a)* the problem can be solved using target tracking. As target vector field we take $G(\theta) = -(L(\theta) - \alpha)$. It is easily seen that $G(\theta)$ is coercive for the optimization problem: if $L(\theta) > \alpha$, the vector field points towards smaller values for θ as it should due to the monotonicity of $L(\theta)$, and for $L(\theta) < \alpha$, the vector field points to larger values, again in correspondence with the monotonicity of $L(\theta)$; moreover $G(\theta)$ is coercive for the optimization problem: if

$L(\theta) > \alpha$, the vector field points towards smaller values for θ as it should due to the monotonicity of $L(\theta)$, and for $L(\theta) < \alpha$, the vector field points to larger values, again in correspondence with the monotonicity of $L(\theta)$; moreover $G(\theta)$ has unique stable point θ^* satisfying $L(\theta^*) = \alpha$. For the stochastic approximation we consider the sequence of the first N interaarival and service times as underlying process, that is,

$$\xi_n(\theta) = (A_1, S_0(\theta), A_2, S_1(\theta), ..., A_N, S_{N-1}(\theta))$$

where we let $S_\theta(\theta) = 0$. We can assume that, given θ_n, the random vector $\xi_n(\theta_n)$ is independent of past values of the underlying process. Using to compute the first N waiting times then leads to the feedback mapping:

$$g(\xi_n(\theta), \theta) = \alpha - \frac{1}{N} \cdot \sum_{k=1}^{N} \mathbb{P}(W_k^{(n)}(\theta_0) < w)(\theta)$$

,

where $W_k^{(n)}$ is the k-th waiting time in the n-th simulation. It is straightforward to see that $G(\theta) = \mathbb{E}[g(\xi_n), \theta)]$ for all n, and the feedback:

$$Y_n = \alpha - \frac{1}{N} \sum_{k=1}^{N} \mathbb{P}(W_k^{(n)}(\theta_n) < w)(\theta_n)$$

To estimate $L(\theta) - \alpha$ we can use $Y_n = \mathbf{1}_{\xi > w} - \alpha$.

(a1) Given $\xi_{n-1} = \xi$, the variable ξ_{n+1} has a mixed distribution with a mass at zero. Call $p_\theta(\xi, \xi')$ the density for $\xi' > 0$. Then we can calculate:

$$p_\theta(\xi, 0) = \mathbb{E}(\mathbb{P}(A_{n+1} > \xi + x | S_n(\theta) = x)) = 1 - \int_0^\infty F_a(\xi + x) f_\theta(x) dx$$

$$p_\theta(\xi, \xi') = \int_0^\infty f_a(x + \xi - \xi') f_\theta(x) dx$$

, for

$$\xi' > 0$$

where f_a and F_a are the density and distribution of the inter-arrival times, respectively. The above transition probability is weakly continuous in (ξ, θ).

The set of stationary measures μ_θ is tight for every compact set Θ in the stability region. Indeed, given a compact set Θ such that $\theta < 1$ for $\theta \in \Theta$, $\underline{\theta} = sup(\theta \in \Theta) < 1$.

$Var(W(\theta)) \leq Var(W(\underline{\theta})) = V < \infty$ by domination argument.

a2 The assumption (a2) is straightforward for $\theta < 1$ in the stability region, because the queue is an ergodic process, so that the stationary measure satisfies:

$$\lim_{N \to \infty} \frac{1}{N} \sum_{n=1}^{N} \mathbf{1}_{\{\xi_n > w\}} = \mathbf{P}(W(\theta) > w) = L(\theta)$$

,

so $\beta_n = 0$ and $g(\xi_n, \theta) = \mathbf{P}(\xi_{n+1} > w|\xi_n)$ calculated using (21) and it is a random variable depending on xi_n.

(a3) and *(a4)* are satisfied for the Y_n of choice.

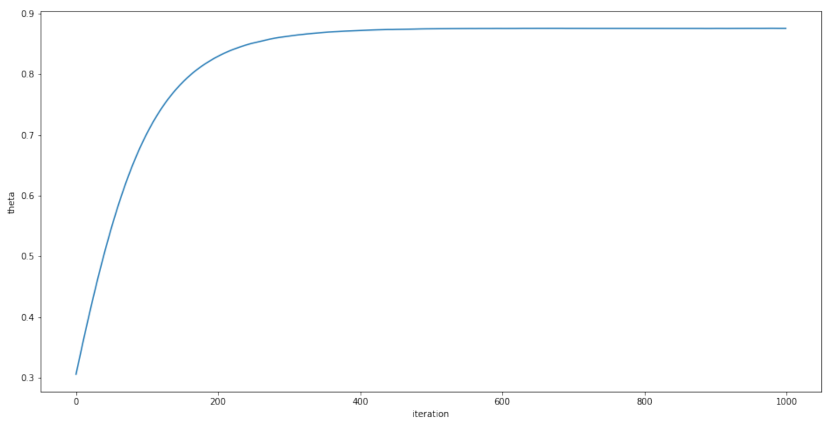
(a5) The ODE:

$$\frac{d\vartheta(t)}{dt} = -(L(\vartheta(t)) - \alpha)$$

,

for $L(\cdot) \in \mathcal{C}^2$ is an increasing function with unique stationary point θ^* which is asymptotically stable point. Our field $G(\theta)$ is coercieve, therefore ODE has bounded trajectories and has a unique limit θ^*

c)



Here we estimate $\mathbb{P}(W_k^{(n)}(\theta_n) < w)(\theta_n)$ using large batch size, that is the reason why the graph looks so smooth and does not depend much on the K. We use this script to generate the plot:

```
import random
import numpy as np
import scipy as sp
import matplotlib.pyplot as plt
%matplotlib inline

np.random.seed(42)
random.seed(42)

def P(theta, w=0.8, std=0.3, samples=10000):
  s = 0
  for i in range(samples):
    rnum = np.random.randn()*std + theta
    if rnum > w:
```

```
        s += 1
    return s / samples

theta = 0.3
alpha = 0.6
N = 5
w = 0.8
epsilon = 0.01
num_iter = 1000
thetas = []

for i in range(num_iter):
    P_est = 0
    for k in range(N):
        P_est = P_est + P(theta, w=0.8)
    theta += epsilon*(alpha - (1/N)*P_est)
    thetas.append(theta)

plt.figure(figsize=(16, 8))
plt.xlabel('iteration')
plt.ylabel('theta')
plt.plot(thetas)
```

Exercise 5.3

a)

Use the stochastic approximation $\theta_{n+1}^{\epsilon} = \theta_n^{\epsilon} + \epsilon Y_n^{\epsilon}$. Then the interpolated process $\vartheta^{\epsilon}(\cdot)$ converge in distribution, as $\epsilon \to 0$, to a limit process which is continious and satisfies the ODE:

$$\frac{d(x(t))}{dt} = G(x(t))$$

.

For the stochastic approximation:

$$\theta_{n+1}^{\epsilon} = \theta_n^{\epsilon} + \epsilon(200 \sum_{k=nK}^{(n+1)K-1} \xi_i - K\alpha)$$

We have $Y_n = (200 \sum_{k=nK}^{(n+1)K-1} \xi_i - K\alpha)$. We assume that ξ_n has distribution as in example 5.4, and assumptions a_1-a_5 are satisfied, then:

$$G(\theta) = \int g(x, \theta)\pi_{\theta}(dx) = \lim_{N\to\infty} \sum_{n=0}^{N} g(\xi_n(\theta), \theta)$$

and for $\{Y_n^{\epsilon}, \epsilon > 0\}$ satisfies

$$g(\xi_{n-1}^{\epsilon}, \theta_{{}_{n}^{\epsilon}}) = \mathbb{E}[Y_n^{\epsilon} | \mathfrak{F}_{n-1}^{\epsilon}, \theta_n^{\epsilon})] = (200 \sum_{k=n}^{Kn} p_{\xi_{n-1},1}(\theta_n) - K\alpha)$$

,

then:

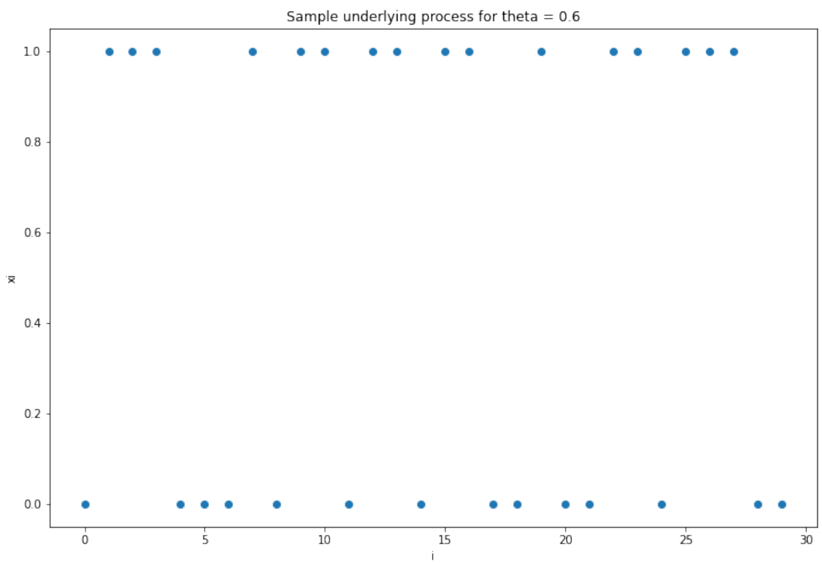
$$\frac{d(\vartheta(t))}{dt} = \lim_{N \to \infty} \sum_{n=0}^{N} (200 \sum_{i=nK}^{(n+1)K-1} \xi_{n-1,1}(\vartheta_n(\theta)) - K\alpha)$$

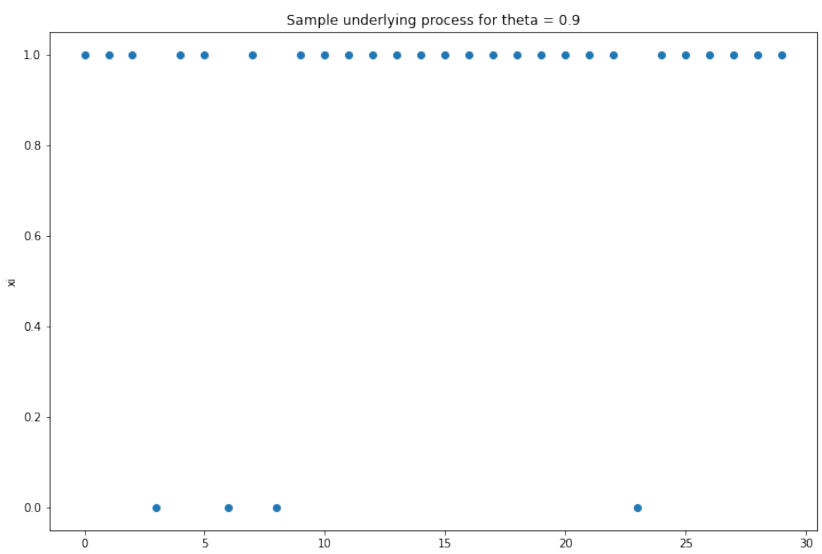
b)

We use $\mathcal{P}_\theta$ from the example 5.4. We use this code to model the markov process:

```
import random
import matplotlib.pyplot as plt

random.seed(42)

class markov_xi(object):

  def __init__(self):
    self.last_state = 0

  def __call__(self, theta):
    p01 = theta
    p11 = theta*theta
    current_state = None
    if self.last_state == 0:
      if random.random() < p01:
        current_state = 1
      else:
        current_state = 0
    else:
        if random.random() < p11:
          current_state = 1
        else:
          current_state = 0
    self.last_state = current_state
    return current_state

def take(generator, n, theta):
  # take n samples from the generator
  for i in range(n):
    yield generator(theta)

def feedback(xi, theta, K, alpha):
  assert K > 0
  return 200*sum(take(xi, n=K, theta=theta)) - K*alpha
```

Here are examples for different θ:

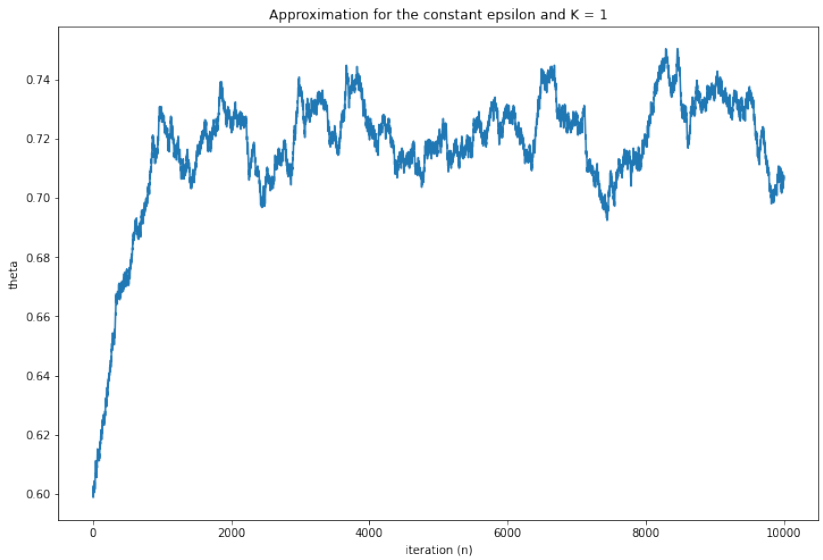


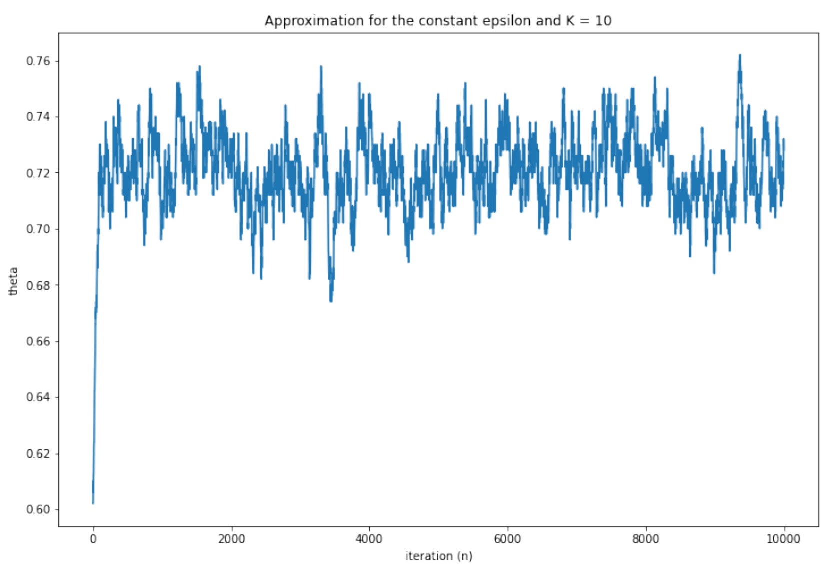


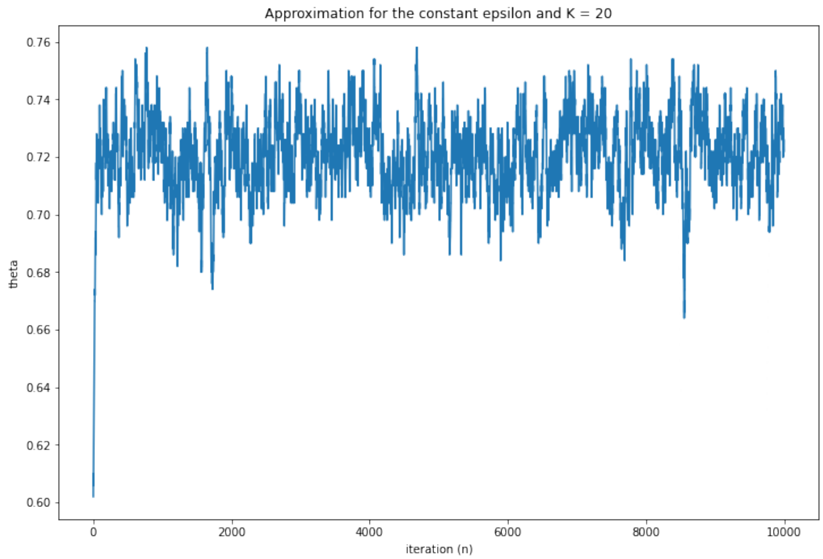

```
epsilon = lambda n: 0.00001

for K in [1, 10, 20]:
    random.seed(42)
    xi = markov_xi()
    thetas = approximate(xi, epsilon, K=K, alpha=120, theta0=0.6, iterations=10000)
    ns = list(range(len(thetas)))

    plt.figure(figsize=(12, 8))
    plt.title('Approximation for the constant epsilon and K = {}'.format(K))
```

```
plt.plot(ns, thetas)
plt.xlabel('iteration (n)')
plt.ylabel('theta')
plt.show()
```

Here we run approximations for different K.



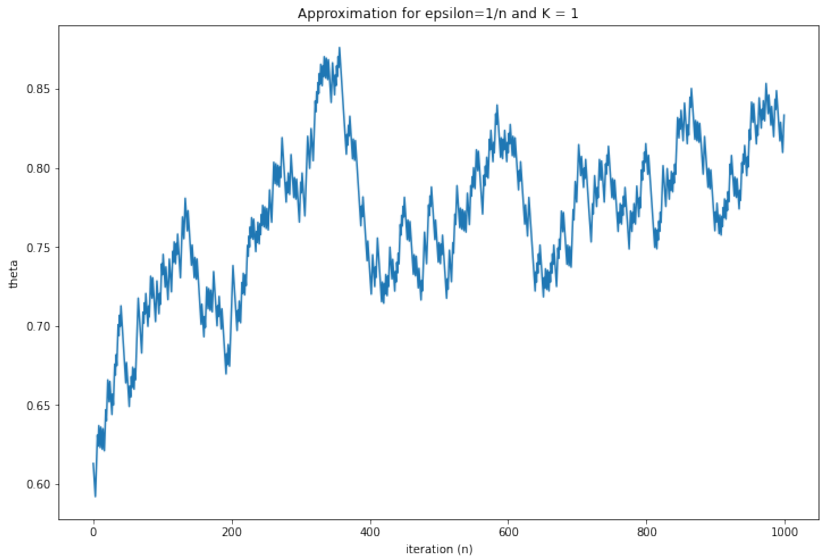


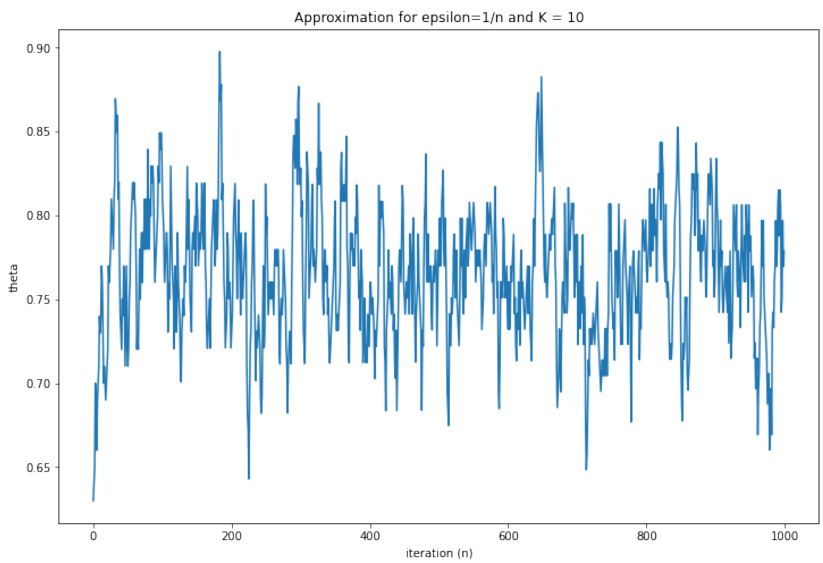


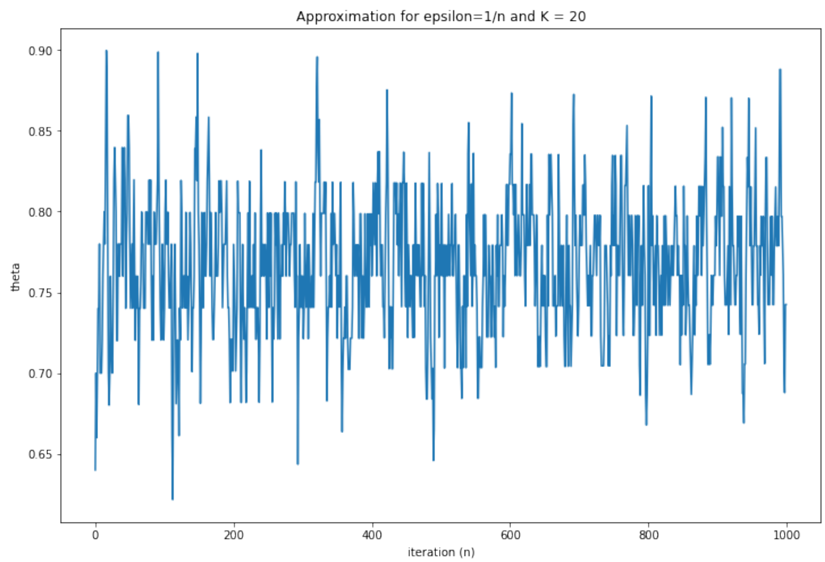
It is clear that the approximation depends on K, but for this particular problem and particular values K it should not make too much of a difference.

c)

```
epsilon = lambda n: 1 / (n+10000.0)

for K in [1, 10, 20]:
  random.seed(42)
  xi = markov_xi()
  thetas = approximate(xi, epsilon, K=K, alpha=130, theta0=0.6, iterations=1000)
  ns = list(range(len(thetas)))

  plt.figure(figsize=(12, 8))
  plt.title('Approximation for epsilon=1/n and K = {}'.format(K))
  plt.plot(ns, thetas)
  plt.xlabel('iteration (n)')
  plt.ylabel('theta')
  plt.show()
```


Approximation for epsilon=1/n and K = 1
theta
iteration (n)
0.60
0.65
0.70
0.75
0.80
0.85
0
200
400
600
800
1000


Approximation for epsilon=1/n and K = 10
theta
iteration (n)
0.65
0.70
0.75
0.80
0.85
0.90
0
200
400
600
800
1000



For an actual product we would choose the case with fixed ϵ because the approximation does not depend on the time and allows the change of the human heartbeat.

6.3

a. Show that $\theta_{n+1} = \theta_n + \epsilon Y_n$, $Y_n = D(\theta_n) - \xi_n$ satisfies the assumptions of Theorem 6.1

(a6)Proof $E[\delta M_n^\epsilon (\delta M_n^\epsilon)^T 1_{||\theta_n^\epsilon - \theta^*|| =< \rho} | \mathfrak{F}_{n-1}^\epsilon]$ **is symmetric matrix**

$$g(\xi_{n-1}^\epsilon, \theta_n^\epsilon) = E[Y_n^\epsilon | \mathfrak{F}_{n-1}^\epsilon]$$

$$\delta M_n^\epsilon = Y_n^\epsilon - g(\xi_{n-1}^\epsilon, \theta_n^\epsilon) = Y_n^\epsilon - E[Y_n^\epsilon | \mathfrak{F}_{n-1}^\epsilon]$$

And $Y_n = \theta_n^{-d} - \xi_n$ Where θ_n is $\mathfrak{F} - \imath$ measurable, and ξ_n is independent. SO:

$$\delta M_n = \theta_n^{-d} - \xi_n - E[\theta_n^{-d} | \mathfrak{F}_{n-1}] + E[\xi_n | \mathfrak{F}_{n-1}]$$

$$= \theta_n^{-d} - \xi_n - \theta_n^{-d} + E[\xi_n]$$

$$= -(\xi_n - E[\xi_n])$$

or

$$\lim_{n \to \infty} \theta_n = \theta^*$$

then $\forall\ \rho > 0$, $\exists\ N > 0$ such that for $n > N$ we have $|\theta_n - \theta^*| < \rho$

then

$$E[\delta M_n^\epsilon (\delta M_n^\epsilon)^T 1_{||\theta_n^\epsilon - \theta^*|| =< \rho} || \mathfrak{F}_{n-1}^\epsilon]$$

$$= E[(\delta M_n^\epsilon)^2|\mathfrak{F}_{n-1}^\epsilon]$$
$$= E[(\xi_n - E[\xi_n])^2|\mathfrak{F}_{n-1}]$$
$$= E[(\xi_n - E[\xi_n])^2] = Var(\xi_n) = 1$$

a7.Where the error term satisfies $E[\rho_1(\theta, \xi_n^\epsilon)] = \mathcal{O}(||\theta - \theta^*||^2)$**, as** $n \to \infty$ $\epsilon \to 0$

$\delta(\theta)$ is known to be analytic and θ^{-d} is infinitely continuously differentiable the analytic on its Domain, then

$$g(\xi_{n-1}^\epsilon, \theta_n^\epsilon) = E[Y_n^\epsilon|\mathfrak{F}_{n-1}^\epsilon]$$
$$= \theta^{-d} - E[\xi_n] = \theta^d - S(\theta) = G$$

G is analytic, so $g(\xi_{n-1}^\epsilon, \theta_n^\epsilon)$ has a Tylor expansion.

a8.There is a Hurwitz matrix A (i.e. a matrix where all the eigenvalues have a negative real part) such that $\lim_{m\to\infty} \frac{1}{m} \sum_{i=n}^{n+m-1} E[\nabla_\theta g(\xi_{n-1}^\epsilon, \theta^*)^T - A] = 0$

Let $A = g'(\theta^*) = D'(theta^*) - S'(\theta^*)$, As we are working in a 1-dimensional space for θ, then

$$\nabla_\theta g^T(\theta^*) = g'(\theta^*)$$
$$\nabla_\theta g^T(\theta^*) - A = 0$$

Which get

$$\lim_{m\to\infty} \frac{1}{m} \sum_{i=n}^{n+m-1} E[\nabla_\theta g(\xi_{n-1}^\epsilon, \theta^*)^T - A] = 0$$

The condition (a1) to (25) from theorem 5.3 hold as they were proved in example 6.3 in our book.

b

Use $d = 5$ **for the demand function. Your economics guru has estimated that** $\theta^* \approx 1$ **and** $S'(\theta^*) \approx 4.5$**. With this information, apply Theorem 6.1 to identify the values of** a**,** σ^2 **for the (approximate) limit Orstein Uhlenbeck process** $U(t)$**, and find** T **such that** $e^{-aT} \approx 0.0001$

$$a = -g'(\theta^*) = -(D'(\theta^*) - S'(\theta^*)) = 5.(1)^{-4} + 4.5 = 9.5$$

because it is 1 dimension, so σ^2 = variance = 1
Because $e^{-aT} \approx 0.0001$ so $T = \frac{log(10^{-4})}{-a} = \frac{log(10^{-4})}{-9.5} \approx 0.9695$

c

Show that $\epsilon \approx 0.0005$ yields a precision of 0.01(half width of the approximate confidence interval after T/ϵ iterations, with confidence level $\alpha = 0.05$).
the asym variance

$$V = \frac{\sigma^2}{2a} = 1/19$$

using a 95% confidence interval
we will get a precision of

$$1.96\sqrt{\frac{0.0005}{19}} \approx 0.01$$

after $N = \frac{0.9695}{0.0005} = 1939$ iterations

d)

In this part of the problem you will generate the random observations ξ_n and run the stochastic approximation. Conditional on θ_n, let $\xi_n \sim LN(m, \upsilon^2)$ have a lognormal distribution. First find the parameters for the m and υ such that (6.10) holds, with $S(\theta) = \theta^s$, $s = 4.3$. Next, run the algorithm and discuss your results.

$$\xi_n \sim LN(m, \upsilon^2)$$

So:

$$E[\xi_n] = exp(u_n + \frac{1}{2}\upsilon_n^2)$$

and

$$var[\xi_n] = exp(2u_n + \upsilon_n^2)[exp(\upsilon_n^2) - 1]$$

Then get:

$$2u_n + \upsilon_n^2 = log(\theta^{2s})$$

$$2u_n + \upsilon_n^2 + log(exp(2u_n + \upsilon_n^2) - 1) = 0$$

Then get:

$$u_n = log(\frac{\theta_n^s}{\sqrt{\theta_n^{-2s} + 1}})$$

$$\upsilon_n^2 = log(\theta_n^{-2s} + 1)$$

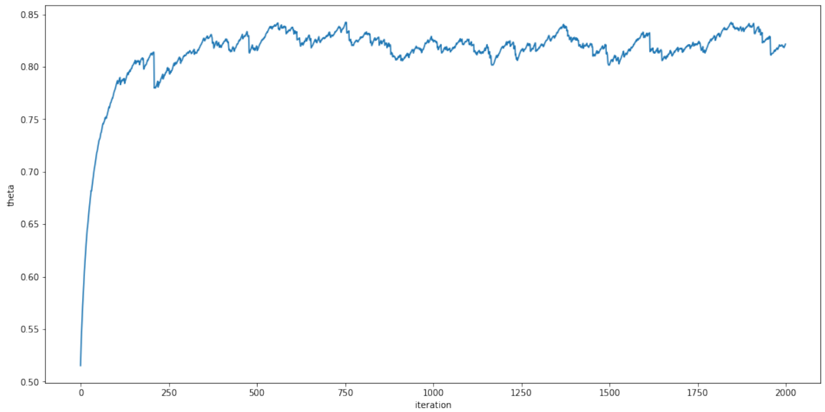
Get:

$$d = 5$$

$$S = 4.3$$

$$\epsilon = 0.0005$$

We choose $\upsilon = 1.0$ and $m = S(\theta)$ and get this plot:



We use this python script to generate the plot:

```
import os
import random
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline

random.seed(42)
np.random.seed(42)

s = 4.3
d = 5

def D(theta):
  return theta**(-d)

def S(theta):
  return theta**s

epsilon = 0.0005
theta = 0.5
thetas = []
for i in range(2000):
  xi = np.random.lognormal(mean=S(theta), sigma=1.0)
  Yn = D(theta) - xi
  theta = theta + epsilon*Yn
  thetas.append(theta)

plt.figure(figsize=(16, 8))
plt.plot(thetas)
plt.ylabel('theta')
plt.xlabel('iteration')
plt.show()
```

Exercise 6.6

a)

$$\mathbb{E}(S_1^2) > \mathbb{E}(X^2(\theta)) > \mathbb{E}(S_2^2(T))$$

$$E(S_1^2(T)) > \mathbb{E}[\theta^2 S_1^2(T) + 2\theta(1-\theta)S_1(T)S_2(T) + (1-\theta)^2 S_2^2(T)] > \mathbb{E}(S_2^2(T))$$

$\theta = 1$: $\mathbb{E}(S_1^2(T)) > \mathbb{E}(S_1^2(T))$ - not true
$\theta = 0$: $\mathbb{E}(S_2^2(T)) > \mathbb{E}(S_2^2(T))$ - not true

b)

$$\phi(x, \theta) = -\theta x_1 - (1-\theta)x_2$$

$$\phi(x, \theta) = -\theta S_1(T) - (1-\theta)S_2(T)$$

$$J(\theta) = \mathbb{E}(\phi(S_1(T), S_2(T); \theta)$$

$$\mathbb{L}(\theta, \lambda) = J(\theta) + \lambda^T g(\theta)$$

$$L(\theta, \lambda) = \mathbb{E}(-\theta S_1(T) - (1-\theta)S_2(T)) + \lambda^T(\mathbb{E}(X^2(\theta)) - B)$$

It is a convex NLP because both $J(\theta)$ and the constraint are convex functions.

c)

d)

e)