# Министерство образования и науки Российской Федерации

## Федеральное государственное бюджетное образовательное учреждение высшего образования

## «Новосибирский государственный технический университет»



## Кафедра теоретической и прикладной информатики

### Лабораторная работа № 1 по дисциплине «Методы активной идентификации динамических систем»

**Оценивание неизвестных параметров**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Факультет: | ПМИ |  |
| Группа: | ПМИМ-01 |
| Студенты: | Ершов П.К., Дорош А.Э. |
| Вариант: | 3 |
| Уровень сложности: | 1 |
| Преподаватель: | Чубич В. М. |

Новосибирск

2021

1. **Цель работы**

Научиться применять метод максимального правдоподобия при оценивании неизвестных параметров моделей линейных дискретных систем.

1. **ЗАДАНИЕ НА ЛАБОРАТОРНУЮ РАБОТУ**

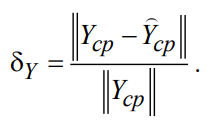
В соответствии с выбранным уровнем сложности:

1. Разработать программы вычисления критерия идентификации и его градиента.
2. Разработать программу нахождения оценок максимального правдоподобия.
3. Следуя своему варианту задания, для указанных истинных значений параметров компьютерным моделированием получить последовательность из 30 измерений, соответствующую указанному входному сигналу. Используя полученные данные наблюдений, вычислить оценки максимального правдоподобия. Для ослабления зависимости результатов оценивания от выборочных данных, осуществить и обработать пять подобных идентификационных экспериментов, запоминая полученные результаты. Усреднив , найти и заполнить следующую таблицу:

|  |  |  |
| --- | --- | --- |
| Номер эксперимента | Значения оценок | Значения оценок |
| 1 |  |  |
| 2 |  |  |
| 3 |  |  |
| 4 |  |  |
| 5 |  |  |
|  |  |  |

Вычислить относительную ошибку оценивания в пространстве параметров по формуле , где - вектор истинных значений параметров.

Отметим, что при решении реальных практических задач истинные значения параметров нам неизвестны. В связи с этим представляется целесообразным вычисление относительной ошибки оценивания в пространстве откликов. Для этого используется соотношение





это усредненные по всем запускам последовательности измерений и их оценок. При этом находится при в соответствии с равенством .

1. **Необходимые теоретические данные**

**Алгоритм вычисления значения критерия идентификации:**

1. Определить F, Ψ , Γ, H , Q , R , , , где

F, Ψ , Γ, H , Q , R – матрицы модели, которые задаются в условиях задачи, – начальный вектор состояний, – ковариационная матрица ошибок прогнозирования.

1. Положить , где

это критерий идентификации, N – число измерений, m – размерность вектора измерений.

1. По формулам (10), (12), (13), (15) найти , , , , где

(10) - ковариационная матрица ошибок одношагового прогнозирования,

(12) ,

(13) ,

(15)

1. Положить .
2. Определить .
3. Положить j = 0.
4. Выбрать , если k = 0.
5. Выбрать , ,, при помощи соотношений (9), (11), (14), где

(9) – оценка одношагового прогнозирования состояния , соответствующая паре ,

(11) ,

(14) - оценка фильтрации состояния , соответствующая паре.

1. Положить , где
2. Увеличить j на единицу. Если , перейти на шаг 7.
3. Увеличить i на единицу. Если , перейти на шаг 5.
4. Положить
5. Увеличить k на единицу. Если , перейти на шаг 3.
6. Положить и закончить процесс.

**Алгоритм вычисления градиентов критерия максимального правдоподобия:**

1. Определить F, Ψ , Γ, H , Q , R , , и

1. Положить

где – градиент критерия идентификации.

1. По формулам (10), (12), (13), (15) найти , , , , где

(10) - ковариационная матрица ошибок одношагового прогнозирования,

(12) ,

(13) ,

(15)

1. По формулам, следующим из (10), (12), (13), (15):

вычислить

1. Положить .
2. Определить .
3. Положить j = 1.
4. Выбрать , если k = 0.
5. Вычислить , ,, при помощи соотношений (9), (11), (14), где
6. По формулам, следующим из (9), (11), (14):

и найти

1. Положить

.

1. Увеличить j на единицу. Если , перейти на шаг 8.
2. Увеличить i на единицу. Если , перейти на шаг 6.
3. Положить .
4. Увеличить k на единицу. Если , перейти на шаг 3, иначе закончить процесс.
5. **Входные данные**

|  |  |  |  |
| --- | --- | --- | --- |
| Матрицы для моделей состояния и измерения | Ковариационные матрицы шумов и начальные условия |  |  |
| ,  ,  , | , R= 0.1,  , | *,*  *,* | , |

1. **Полученные результаты и их анализ**

*Результаты работы алгоритма вычисления значения критерия идентификации:*

|  |  |  |
| --- | --- | --- |
| Номер эксперимента | Значения оценок | Значения оценок |
| 1 | -1.4996 | 0.8000 |
| 2 | -1.5011 | 0.3601 |
| 3 | -1.5011 | 0.1913 |
| 4 | -1.4993 | 0.8000 |
| 5 | -1.4992 | 0.4629 |
|  | -1.5001 | 0.5229 |

Относительная ошибка оценивания = 0.0145.

Относительная ошибка оценивания = 5.0292e-04.

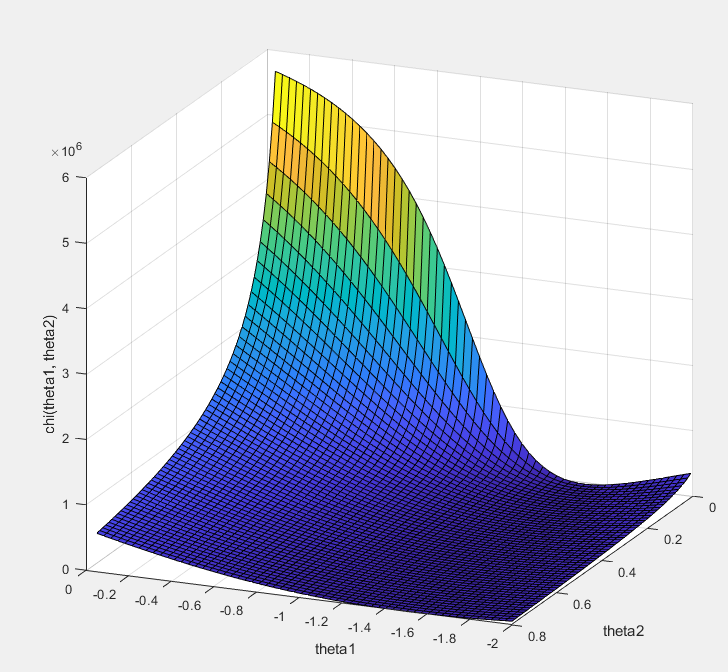
*Результаты работы алгоритма вычисления градиентов критерия максимального правдоподобия:*

|  |  |  |
| --- | --- | --- |
| Номер эксперимента | Значения оценок | Значения оценок |
| 1 | -1.4996 | 0.8000 |
| 2 | -1.5011 | 0.3601 |
| 3 | -1.5011 | 0.1913 |
| 4 | -1.4993 | 0.8000 |
| 5 | -1.4992 | 0.4629 |
|  | -1.5001 | 0.5229 |

Относительная ошибка оценивания = 0.0145.

Относительная ошибка оценивания = 5.0292e-04.

1. **График оценок по и**



1. **Текст программы**

Файл initial\_data.m

function [F, Psi, Gamma, H, Q, R, x\_0, P\_0, u, x, y] = initial\_data(theta, N)

F = [-0.8, 1; theta(1), 0];

Psi = [1; 1];

Gamma = [1; 1];

H = [1, 0];

Q = theta(2);

R = 0.1;

x\_0 = [0; 0];

P\_0 = [0.1, 0; 0, 0.1];

u = 3;

w = sqrt(Q) \* randn(30, 1);

v = sqrt(R) \* randn(30, 1);

x = cell(30, 1);

y = zeros(30, 1);

x(1) = {F \* x\_0 + Psi \* u + Gamma \* w(1)};

y(1) = H \* x{1} + v(1);

for k = 1 : 29

x(k+1) = {F \* x{k} + Psi \* u + Gamma \* w(k+1)};

y(k+1) = H \* x{k+1} + v(k+1);

end

end

Файл algorithm\_1.m

function [Hi] = algorithm\_1(theta, y)

[F, Psi, Gamma, H, Q, R, x\_0, P\_0, u, ~, ~] = initial\_data (theta, 30);

m = 1;

v = 1;

N = length(y);

I = [1, 0;

0, 1];

Hi = N \* m \* log(2 \* pi);

P\_k\_k = P\_0;

x\_k\_k = x\_0;

y\_t = zeros(1,N); % для подсчета нормы

for k = 0 : N - 1

P\_k1\_k = F \* P\_k\_k \* F' + Gamma \* Q \* Gamma';

B\_k1 = H \* P\_k1\_k \* H' + R;

K\_k1 = P\_k1\_k \* H' / B\_k1;

P\_k1\_k1 = (I - K\_k1 \* H )\* P\_k1\_k;

delta = 0;

for i = 1 : 1

for j = 1 : 1

x\_k1\_k = F \* x\_k\_k + Psi \* u;

eps = y(k+1) - H \* x\_k1\_k;

x\_k1\_k1 = x\_k1\_k + K\_k1 \* eps;

y\_t(k+1) = H \* x\_k1\_k1;

delta = delta + eps' / B\_k1 \* eps;

end

end

Hi = Hi + v \* log(det(abs(B\_k1))) + delta;

P\_k\_k = P\_k1\_k1;

x\_k\_k = x\_k1\_k1;

end

Hi = Hi / 2;

%normaY = norm(y - y\_t')/norm(y)

end

Файл initial\_gradient.m

function [F\_grad, Psi\_grad, Gamma\_grad, H\_grad, Q\_grad, R\_grad, x0\_grad, P0\_grad] = initial\_gradient(theta)

F\_grad = cell(2,1);

F\_grad{1} = [0, 0; 1, 0];

F\_grad{2} = [0, 0; 0, 0];

Psi\_grad = cell(2,1);

Psi\_grad{1} = [0;0];

Psi\_grad{2} = [0;0];

Gamma\_grad = cell(2,1);

Gamma\_grad{1} = [0;0];

Gamma\_grad{2} = [0;0];

H\_grad = cell(2,1);

H\_grad{1} = [0,0];

H\_grad{2} = [0,0];

Q\_grad = cell(2,1);

Q\_grad{1} = 0;

Q\_grad{2} = 1;

R\_grad = cell(2,1);

R\_grad{1} = 0;

R\_grad{2} = 0;

x0\_grad = cell(2,1);

x0\_grad{1} = [0;0];

x0\_grad{2} = [0;0];

P0\_grad = cell(2,1);

P0\_grad{1} = [0, 0; 0, 0];

P0\_grad{2} = [0, 0; 0, 0];

end

Файл algorithm\_2.m

function [Hi, Hi\_grad] = algorithm\_2(theta, y)

N = length(y);

alpha = length(theta);

m = 1;

v = 1;

I = [1, 0;

0, 1];

[F, Psi, Gamma, H, Q, R, x0, P0, u, ~, ~] = initial\_data(theta);

[F\_grad, Psi\_grad, Gamma\_grad, H\_grad, Q\_grad, R\_grad, x0\_grad, P0\_grad] = initial\_gradient(theta);

Hi = N \* m \* v \* log(2 \* pi);

Hi\_grad = cell(alpha, 1);

y\_t = zeros(1,N); % для подсчета нормы

for i = 1 : alpha

Hi\_grad{i} = 0;

end

P\_k\_k = P0;

P\_k\_k\_grad = P0\_grad;

P\_k1\_k\_grad = cell(alpha, 1);

B\_grad = cell(alpha, 1);

K\_grad = cell(alpha, 1);

P\_k1\_k1\_grad = cell(alpha, 1);

x\_k1\_k\_grad = cell(alpha, 1);

e\_grad = cell(alpha, 1);

x\_k1\_k1\_grad = cell(alpha, 1);

for k = 0 : N - 1

P\_k1\_k = F \* P\_k\_k \* F' + Gamma \* Q \* Gamma';

B\_k1 = H \* P\_k1\_k \* H' + R;

K\_k1 = P\_k1\_k \* H' / B\_k1;

P\_k1\_k1 = (I- K\_k1 \* H )\* P\_k1\_k;

for a = 1 : alpha

P\_k1\_k\_grad{a} = F\_grad{a} \* P\_k\_k \* F' + F \* P\_k\_k\_grad{a} \* F'+ F \* P\_k\_k \* F\_grad{a}' + Gamma\_grad{a} \* Q \* Gamma' + Gamma \* Q\_grad{a} \* Gamma' + Gamma \* Q \* Gamma\_grad{a}';

B\_grad{a} = H\_grad{a} \* P\_k1\_k \* H' + H \* P\_k1\_k\_grad{a} \* H' + H \* P\_k1\_k \* H\_grad{a}' + R\_grad{a};

K\_grad{a} = (P\_k1\_k\_grad{a} \* H' + P\_k1\_k \* H\_grad{a}' - P\_k1\_k \* H' / B\_k1 \* B\_grad{a}) / B\_k1;

P\_k1\_k1\_grad{a} = (I - K\_k1 \* H) \* P\_k1\_k\_grad{a} - (K\_grad{a} \* H + K\_k1 \* H\_grad{a}) \* P\_k1\_k;

end

delta = 0;

delta\_grad = cell(alpha, 1);

for a = 1 : alpha

delta\_grad(a) = {0};

end

for i = 1 : 1 % шаг 5

for j = 1 : 1

if (k == 0) % шаг 8

x\_k\_k = x0;

x\_k\_k\_grad = x0\_grad;

end

x\_k1\_k = F \* x\_k\_k + Psi \* u; % 9

eps = y(k+1) - H \* x\_k1\_k; % 11

x\_k1\_k1 = x\_k1\_k + K\_k1 \* eps; % 14

y\_t(k+1) = H \* x\_k1\_k1;

delta = delta + eps' / B\_k1 \* eps; % для хи

for a = 1 : alpha % 10 шаг

x\_k1\_k\_grad{a} = F\_grad{a} \* x\_k\_k + F \* x\_k\_k\_grad{a} + Psi\_grad{a} \* u;

e\_grad{a} = - H\_grad{a} \* x\_k1\_k - H \* x\_k1\_k\_grad{a};

x\_k1\_k1\_grad{a} = x\_k1\_k\_grad{a} + K\_grad{a} \* eps + K\_k1 \* e\_grad{a};

end

for a = 1 : alpha

delta\_grad{a} = delta\_grad{a} + e\_grad{a}' / B\_k1 \* eps - 0.5 \* eps' / B\_k1 \* B\_grad{a} / B\_k1 \* eps;

end

end

end

for a = 1 : alpha

Hi\_grad{a} = Hi\_grad{a} + v / 2 \* trace(B\_k1 \ B\_grad{a}) + delta\_grad{a};

end

P\_k\_k = P\_k1\_k1;

P\_k\_k\_grad = P\_k1\_k1\_grad;

x\_k\_k = x\_k1\_k1;

x\_k\_k\_grad = x\_k1\_k1\_grad;

Hi = Hi + v \* log(det(abs(B\_k1))) + delta; % вне алгоритма градиента, но сама функция нужна

end

Hi = Hi / 2;

Hi\_grad = [Hi\_grad{1} ; Hi\_grad{2}];

norma\_grad\_Y = norm(y - y\_t')/norm(y);

end

Файл main.m

N = 30;

t\_n = 5;

theta\_ist = [-1.5; 0.5];

theta\_1 = zeros(2,t\_n);

theta\_2 = zeros(2,t\_n);

rng(0, 'twister');

for i = 1 : t\_n

[ ~, ~, ~, ~, ~, ~, ~, ~, ~, X, Y] = initial\_data(theta\_ist, N);

opt = optimoptions('fmincon', 'Algorithm', 'interior-point');

fun = @(x) algorithm\_1(x, Y);

theta\_1(:,i) = fmincon(fun, [-1.25; 0.2], [], [], [], [], [-2; 0.01], [-0.05; 0.8], [], opt);

opt = optimoptions('fmincon', 'Algorithm', 'interior-point', 'SpecifyObjectiveGradient',true, 'GradObj','on');

fun\_grad = @(x) algorithm\_2(x, Y);

theta\_2(:,i) = fmincon(fun\_grad, [-1.25; 0.2], [], [], [], [], [-2; 0.01], [-0.05; 0.8], [], opt);

end

disp(theta\_1)

disp(theta\_2)

m\_1 = mean(theta\_1, 2)

m\_2 = mean(theta\_2, 2)

theta\_disp1 = norm(theta\_ist - m\_1) / norm(theta\_ist)

theta\_disp2 = norm(theta\_ist - m\_2) / norm(theta\_ist)

f = @(x,y) algorithm\_2([x; y], Y);

xx = linspace(-2, -0.05, 60);

yy = linspace(0.01, 0.8, 60);

a = zeros(length(xx), length(yy));

for i = 1 : length(xx)

for j = 1 : length(yy)

a(j, i) = f(xx(i), yy(j));

end

end

surf(xx, yy, a);

hold on

xlabel('theta1'), ylabel('theta2'), zlabel('chi(theta1, theta2)');