

## 機械学習教師付き学習レポート課題

### SVM の実装

チャンヴァンサン

0 5 — 1 6 1 0 2 3

実装コード

#language: GNU Octave

#SVM with backtracking line search

clear all;

#gendata

rand('state',0);

randn('state',0);

global n = 200;

a = linspace(0,4\*pi,n/2);

u = [a.\*cos(a) (a+pi).\*cos(a)]'+rand(n,1);

v = [a.\*sin(a) (a+pi).\*sin(a)]'+rand(n,1);

global x = [u v];

global y = [ones(1,n/2) -ones(1,n/2)]';

#kernel

#configuration

global step = 1;

global eps = 1;

global h = 0.7;

global C = 0.5;

global hh = 2\*h^2;

#backtracking line search

global alpha = 0.5;

global beta = 0.8;

function retval = gauss(u, v)

global hh;

w = u - v;

retval = exp(- w \* w' / hh);

endfunction

function retval = calc\_K()

global n;

global x;

retval = zeros(n, n);

```

        for i = 1:n
            for j = 1:i
                retval(i, j) = gauss(x(i, :), x(j, :));
                retval(j, i) = retval(i, j);
            endfor
        endfor
    endfunction

```

```

global K = calc_K(x, n, hh);
global K2 = 2 * K;

```

```

function retval = f(theta)
    global C;
    global K;
    global y;
    t = 1 - (K * theta) .* y;
    t(t < 0) = 0;
    retval = C * sum(t) + theta' * K * theta;
endfunction

```

```

global K_signed = K .* repmat(y, 1, n);

```

```

function retval = delta(theta)
    global K_signed;
    global K;
    global y;
    s = sign(1 - (K * theta) .* y);
    s(s < 0) = 0;
    retval = -(s' * K_signed)';
endfunction;

```

```

function retval = nabla(theta)
    global C;
    global K2;
    retval = C * delta(theta) + K2 * theta;
endfunction

```

```

function retval = next_theta(theta)
    ok = false;
    global step;
    global alpha;
    global beta;
    st = step;
    i = 0;
    threshold = 10;
    do

```

```

        i = i + 1;
        nl = nabla(theta);
        if (f(theta - st * nl) - f(theta) > -alpha * st * (theta' * theta))
            st = beta * st;
        else
            ok = true;
        endif
    until (ok || i > threshold)
        retval = theta - st * nl;
endfunction;

```

```

function retval = SVM()
    global n;
    retval = randn(n, 1);
    #retval = zeros(n, 1);
    i = 0;
    threshold = 50;
    do
        i = i + 1;
        last = retval;
        retval = next_theta(retval);
        diff = retval - last;
        #norm(retval - last)
    until (abs(norm(retval - last)) < eps || i > threshold);
endfunction;

```

```

t = SVM();

```

```

#draw
m = 100;
X = linspace(-15,15,m)';
X2 = X.^2;

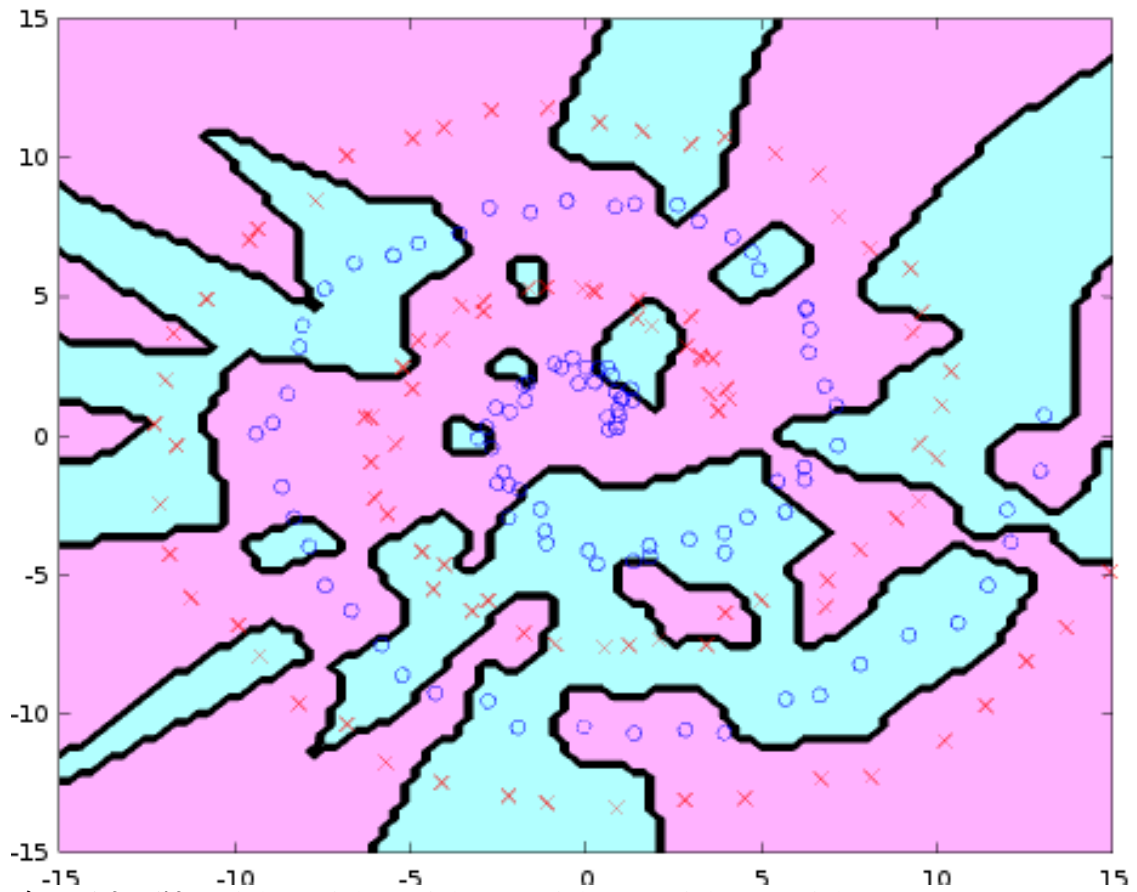
```

```

U = exp(-(repmat(u.^2,1,m)+repmat(X2',n,1)-2*u*X')/hh);
V = exp(-(repmat(v.^2,1,m)+repmat(X2',n,1)-2*v*X')/hh);
figure(1);
clf;
hold on;
axis([-15 15 -15 15]);
contourf(X,X,sign(V.*(U.*repmat(t,1,m))));
plot(x(y == 1,1),x(y == 1,2),'bo');
plot(x(y == -1,1),x(y == -1,2),'rx');
colormap([1 0.7 1; 0.7 1 1]);

```

実行結果：  
初期値（ランダム）



次はそれぞれ1回、2回、3回、10回、20回、50回 backtracking line search を繰り返した結果：

