

```

1 Inputs:
2   y: array of {+1, -1}: class of the i-th instance
3   Q:  $Q[i][j] = y[i]y[j]K[i][j]$ ; K: kernel matrix
4   len: number of instances
5
6 //parameters
7 eps = 1e-3 // stopping tolerance
8 tau = 1e-12
9
10 //main routine
11 initialize alpha array A to all zero
12 initialize gradient array G to all -1
13
14 while(1)
15 {
16   (i,j) = selectB()
17   if (j == -1)
18     break
19
20   //working set is (i,j)
21   a =  $Q[i][i] + Q[j][j] - 2y[i]y[j]Q[i][j]$ 
22   if (a <= 0)
23     a = tau
24   b =  $-y[i]G[i] + y[j]G[j]$ 
25
26   //update alpha
27   oldAi = A[i], oldAj = A[j]
28   A[i] += y[i]*b/a
29   A[j] -= y[j]*b/a
30
31   //project alpha back to the feasible region
32   sum = y[i]*oldAi + y[j]*oldAj
33   if A[i] > C
34     A[i] = C
35   if A[i] < 0
36     A[i] = 0
37   A[j] = y[j]*(sum - y[i]*A[i])
38
39   if A[j] > C
40     A[j] = C
41   if A[j] < 0
42     A[j] = 0
43   A[i] = y[i]*(sum - y[j]*A[j])
44
45   //update gradient
46   deltaAi = A[i] - oldAi, deltaAj = A[j] - oldAj
47   for t = 1 to len
48     G[t] += Q[t][i]*deltaAi + Q[t][j]*deltaAj
49 }
50
51 procedure selectB
52   //select i
53   i = -1
54   G_max = -inf
55   G_min = inf
56   for t = 1 to len
57     if (y[t]==+1 and A[t] < C) or (y[t]==-1 and A[t] > 0)
58     {
59       if (-y[t]*G[t] >= G_max)
60       {
61         i = t

```

```

62         G_max = -y[t]*G[t]
63     }
64 }
65
66 //select j
67 j = -1
68 obj_min = inf
69 for t = 1 to len
70 {
71     if(y[t]==+1 and A[t] >0)or(y[t]==-1 and A[t] < C)
72     {
73         b = G_max + y[t]*G[t]
74         if (-y[t]*G[t] <= G_min)
75             G_min = -y[t]*G[t]
76         if (b > 0)
77         {
78             a = Q[i][i]+Q[t][t]-2*y[i]*y[t]*Q[i][t]
79             if (a <= 0)
80                 a = tau
81             if (-(b*b)/a <= obj_min)
82             {
83                 j = t
84                 obj_min = -(b*b)/a
85             }
86         }
87     }
88 }
89
90 if (G_max-G_min < eps)
91     return (-1,-1)
92
93 return (i,j)
94 end procedure

```