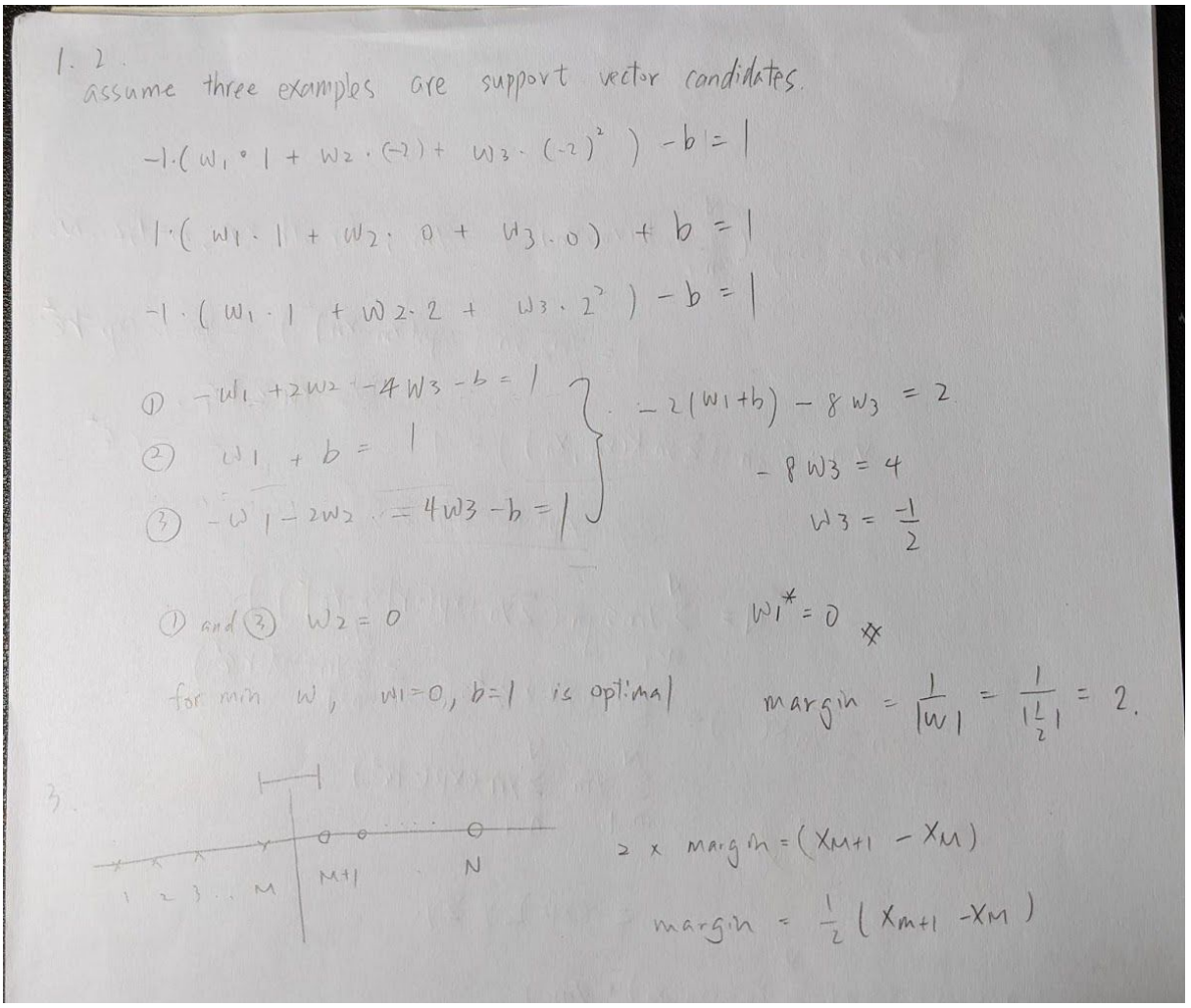
1. d
2. b
3. e

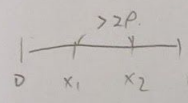

1. 2.
assume three examples are support vector candidates.
$-1\cdot(w_1\cdot 1 + w_2\cdot(-2) + w_3\cdot(-2)^2) - b = 1$
$1\cdot(w_1\cdot 1 + w_2\cdot 0 + w_3\cdot 0) + b = 1$
$-1\cdot(w_1\cdot 1 + w_2\cdot 2 + w_3\cdot 2^2) - b = 1$
① $-w_1 + 2w_2 - 4w_3 - b = 1$
② $w_1 + b = 1$
③ $-w_1 - 2w_2 - 4w_3 - b = 1$
$-2(w_1+b) - 8w_3 = 2$
$-8w_3 = 4$
$w_3 = -\frac{1}{2}$
① and ③ $w_2 = 0$
$w_1^* = 0$ ※
for min w, $w_1=0$, $b=1$ is optimal
$\text{margin} = \frac{1}{|w|} = \frac{1}{|-\frac{1}{2}|} = 2.$
3.
1 2 3 . . M
M+1
N
$2 \times \text{margin} = (x_{M+1} - x_M)$
$\text{margin} = \frac{1}{2}(x_{M+1} - x_M)$

4. a

4.

if x_1, x_2 are both with same sign, the expected dictomomies are 2.

So we just consider the other two cases with different signs.



, $\rho \in [0, 0.5]$, $|x_2 - x_1| > 2\rho$.

$$\int_0^{1-2\rho} \int_{x_1+2\rho}^{1} dx_2 \, dx_1 = \int_0^{1-2\rho} 1-(2\rho+x_1)\, dx_1 \text{ , if } x_1+2\rho < 1 \text{ then all } > 0.$$

$$= \int_0^{1-2\rho} (1-2\rho)-x_1 \, dx_1 = (1-2\rho)\Big|_0^{1-2\rho} - \frac{1}{2}(1-2\rho)^2 = \frac{1}{2}$$

$2 \cdot \left(\frac{1}{2}(1-2\rho)^2\right) = (1-2\rho)^2$, for $x_2 > x_1$, same as $x_1 > x_2$.

$\therefore$ overall, $2 + 2 \cdot (1-2\rho)^2$ ✱

5. c
6. e

3.

$$\mathcal{L}(b,w,\alpha) = \frac{1}{2}w^Tw + \sum[y_n=1]\alpha_n(\rho^+ - y_n(w^Tx_n+b)) + \sum[y_n=-1]\alpha_n(\rho^- - y_n(w^Tx_n+b))$$

$$\frac{\partial}{\partial b}=0,\quad \text{problem} = \max\left(\min \frac{1}{2}w^Tw + \sum[y_n=1]\alpha_n(\rho^+ - y_nw^Tx_n) + \sum[y_n=-1]\alpha_n(\rho^- - y_nw^Tx_n)\right)$$

$$\frac{d}{dw_i}=0,\quad w = \sum\alpha_n[y_n=1]x_n + \sum\alpha_n[y_n=-1]y_n = \sum\alpha_ny_nx_n.$$

$$\text{problem} = \max\left(\sum\rho^+[y_n=1]\alpha_n + \sum\rho^-[y_n=-1]\alpha_n - \frac{1}{2}w^Tw\right)$$

$$\downarrow \times(-1)$$

$$= \min \frac{1}{2}w^Tw - \left(\sum_{n=1}^{N}\rho_+[y_n=1]\alpha_n + \sum_{n=1}^{N}\rho_-[y_n=-1]\alpha_n\right) \quad ※$$

6. for original hard margin, $\sum\alpha_n^*y_n=0$

uneven-margin, $\sum\alpha_ny_n=0$

$$\sum\alpha_n^*y_n = \sum_{y_n=+1}\alpha_n^* - \sum_{y_n=-1}\alpha_n^* = 0;\quad \sum_{y_n=1}\alpha_n - \sum_{y_n=-1}\alpha_n = 0,\quad \sum_{y_n=1}\alpha_n = \sum_{y_n=-1}\alpha_n$$

$$\rho_+\sum_{y_n=1}\alpha_n + \rho_-\sum_{y_n=-1}\alpha_n = (\rho_+ + \rho_-)\sum_{y_n=+1}\alpha_n = (\rho_+ + \rho_-)\sum_{y_n=-1}\alpha_n = \left(\frac{\rho_+ + \rho_-}{2}\right)\sum_{n=1}^{N}\alpha_n$$

Under gradient constraint of optimal solution

$Q\alpha + \rho = 0$, for un-even case

$Q\alpha^* + 1 = 0$, for original case.

$$\downarrow$$

$Q\left(\frac{\rho_+ + \rho_-}{2}\right)\alpha^* + \frac{(\rho_+ + \rho_-)}{2}1 = 0$ is also a solution!

$$\frac{\rho_+ + \rho_-}{2}\alpha^*$$

7. d
8. c
9. d
10. c

7.

d. $\log_2 K(x,x')$ may less than 0 when $0 < K(x,x') < 1$,
which may cause the element of matrix non-positive
not positive definite matrix ZZ^T → not valid kernel ※

8.

$$\|\phi(x) - \phi(x')\|^2 = \phi(x)^T\phi(x) - 2\phi(x)^T\phi(x') + \phi(x')^T\phi(x')$$
$$= K(x,x) + K(x',x') - 2K(x,x')$$
$$= 1 + 1 - 2K(x,x') \leq 2 \text{ ※}$$

10.

$$w_{t+1} = \sum_{n=1}^{N} \alpha_{t+1}\phi(x_n) = w_t + y_{n(t)}\phi(x_{n(t)}) = \left(\sum_{n=1}^{N} \alpha_t \phi(x_n)\right) + y_{n(t)}\phi(x_n)$$

Compare coefficient of each $\phi(x)$, $\alpha_{t+1} = \alpha_t$ except $\phi(x_n)$, whose coefficient is $\alpha_{t,n(t)} + y_n(t)$ ※

9. problem can be viewed as:

$$h(x_m) = \text{sign}\left(\sum_{n=1}^{N} y_n K(x_m, x_n)\right) = y_m, \forall m = 1,2..N, E_{in} = 0$$

$$\sum_{n=1}^{N} y_n k(x_m, x_n) = \sum_{\substack{n=1 \\ n\neq m}}^{N} y_n \exp(-r(x_m - x_n)^2) + y_m.$$

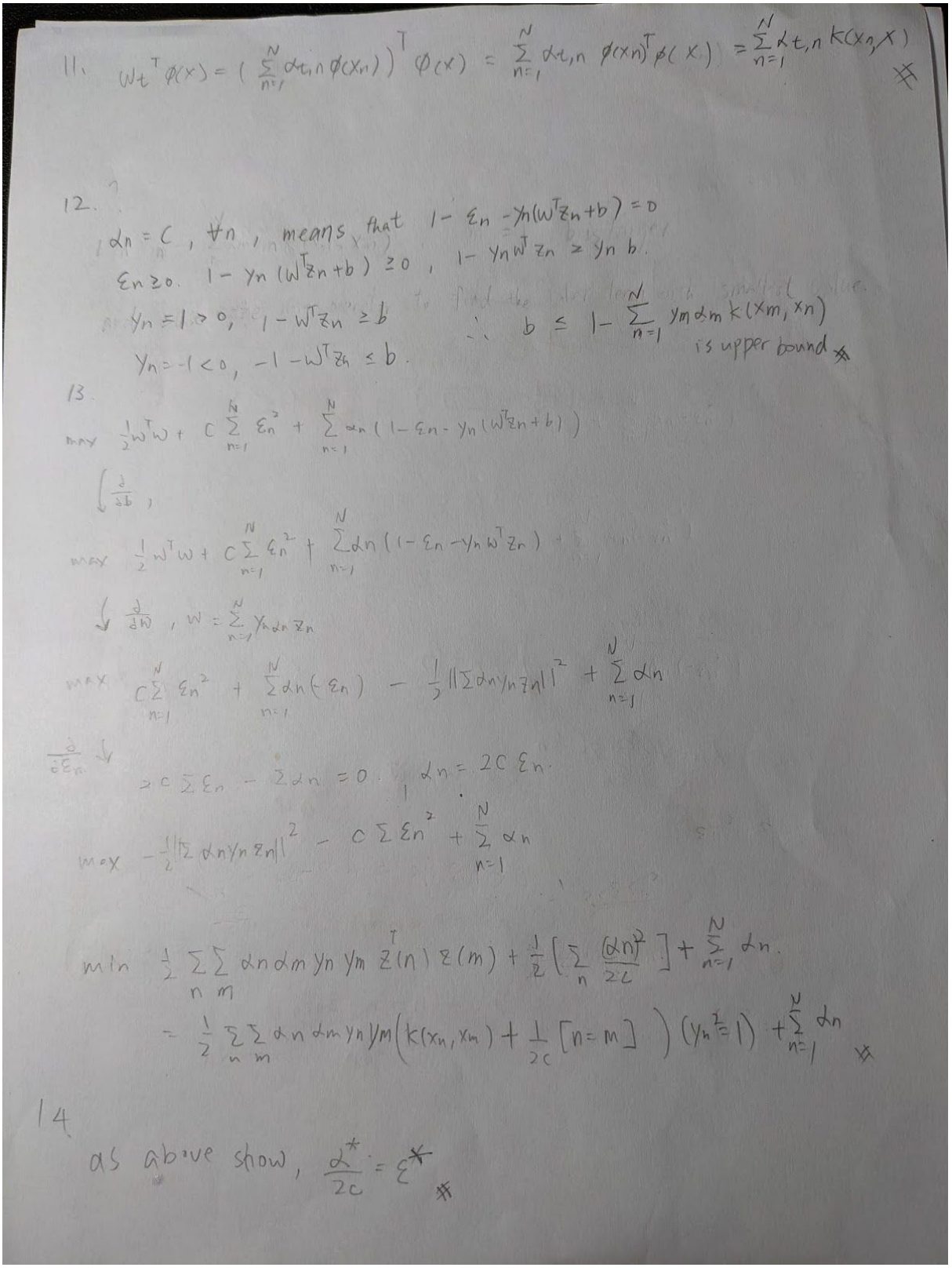
$$\left|\sum_{\substack{n\neq m \\ n=1}}^{N} y_n \exp(-r(x_m - x_n)^2)\right| < 1, \text{ will maintain constraint.}$$

$$-1 < \sum y_n \exp(-k(x_m - x_n)^2) \leq \sum y_n \exp(-r\varepsilon^2) < 1$$

$$-1 < -(N-1)\exp(-r\varepsilon^2) < \sum y_n \exp(-r\varepsilon^2) < (N-1)\exp(-r\varepsilon^2) < 1, \text{ if } r \text{ is large enough.}$$

$\therefore \exp(-r\varepsilon^2) < \frac{1}{N-1}$ for both side $-r\varepsilon^2 < \ln(\frac{1}{N-1})$ $r > \frac{\ln(N-1)}{\varepsilon^2}$ ※

11. a
12. b
13. e
14. e



11. $w_t^T \phi(x) = \left(\sum_{n=1}^{N} \alpha_{t,n} \phi(x_n)\right)^T \phi(x) = \sum_{n=1}^{N} \alpha_{t,n} \phi(x_n)^T \phi(x) = \sum_{n=1}^{N} \alpha_{t,n} k(x_n, x)$ #

12.

$\alpha_n = C$, $\forall n$, means that $1 - \xi_n - y_n(w^T z_n + b) = 0$

$\xi_n \geq 0$. $1 - y_n(w^T z_n + b) \geq 0$, $1 - y_n w^T z_n \geq y_n b$.

$y_n = 1 > 0$, $1 - w^T z_n \geq b$

$y_n = -1 < 0$, $-1 - w^T z_n \leq b$.

$\therefore b \leq 1 - \sum_{n=1}^{N} y_m \alpha_m k(x_m, x_n)$ is upper bound #

13.

$$\max \frac{1}{2} w^T w + C \sum_{n=1}^{N} \xi_n^2 + \sum_{n=1}^{N} \alpha_n (1 - \xi_n - y_n(w^T z_n + b))$$

$\downarrow \frac{\partial}{\partial b}$,

$$\max \frac{1}{2} w^T w + C \sum_{n=1}^{N} \xi_n^2 + \sum_{n=1}^{N} \alpha_n (1 - \xi_n - y_n w^T z_n)$$

$\downarrow \frac{\partial}{\partial w}$, $w = \sum_{n=1}^{N} y_n \alpha_n z_n$

$$\max C \sum_{n=1}^{N} \xi_n^2 + \sum_{n=1}^{N} \alpha_n (-\xi_n) - \frac{1}{2} \left\| \sum \alpha_n y_n z_n \right\|^2 + \sum_{n=1}^{N} \alpha_n$$

$\downarrow \frac{\partial}{\partial \xi_n}$

$2C \sum \xi_n - \sum \alpha_n = 0$, $\alpha_n = 2C \xi_n$.

$$\max -\frac{1}{2} \left\| \sum \alpha_n y_n z_n \right\|^2 - C \sum \xi_n^2 + \sum_{n=1}^{N} \alpha_n$$

$$\min \frac{1}{2} \sum_n \sum_m \alpha_n \alpha_m y_n y_m z(n)^T z(m) + \frac{1}{2} \left[\sum_n \frac{(\alpha_n)^2}{2C} \right] + \sum_{n=1}^{N} \alpha_n.$$

$$= \frac{1}{2} \sum_n \sum_m \alpha_n \alpha_m y_n y_m \left(k(x_n, x_m) + \frac{1}{2C} [n = m] \right) (y_n^2 = 1) + \sum_{n=1}^{N} \alpha_n$$ #

14.

as above show, $\frac{\alpha^*}{2C} = \xi^*$ #

15. d

a. result, 8.457084298367683

```
yiwenlai@YiWens-MBP ~/Desktop/html_hw5/libsvm-3.24/python INSERT python3 train.py
.....*..*.....*
optimization finished, #iter = 22874
nu = 0.108695
obj = -4784.881503, rho = 3.623003
nSV = 500, nBSV = 466
Total nSV = 500
Accuracy = 95.8061% (4249/4435) (classification)
8.457084298367683
```

b. Code

```
y, x = svm_read_problem('hw5_train')
for i in range(len(y)):
    if y[i] == 3.0:
        y[i] = 1
    else:
        y[i] = -1
m = svm_train(y, x, '-c 10 -s 0 -t 0')
p_label, p_acc, p_val = svm_predict(y, x, m)
support_vectors = m.get_SV()
support_vector_coefficients = m.get_sv_coef()
w = []
for i in range(36):
    wi = 0
    for j in range(len(support_vector_coefficients)):
        if i+1 in support_vectors[j]:
            wi +=
support_vector_coefficients[j][0]*support_vectors[j]
[i+1]
    w.append(wi)
w_2 = [x*x for x in w]
print(math.sqrt(sum(w_2)))
```

16. b

a. result is presented in Q17

17. c

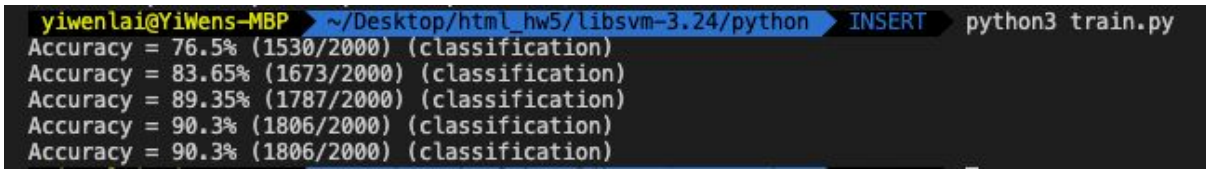
a. result for Q16,17

```
yiwenlai@YiWens-MBP ~/Desktop/html_hw5/libsvm-3.24/python INSERT python3 train.py
Class:  1  Number of support vectors:  145  Accuracy:  99.93235625704622
Class:  2  Number of support vectors:  87  Accuracy:  100.0
Class:  3  Number of support vectors:  433  Accuracy:  97.76775648252537
Class:  4  Number of support vectors:  712  Accuracy:  95.98647125140924
Class:  5  Number of support vectors:  259  Accuracy:  99.32356257046223
```

b. Code for Q16,17

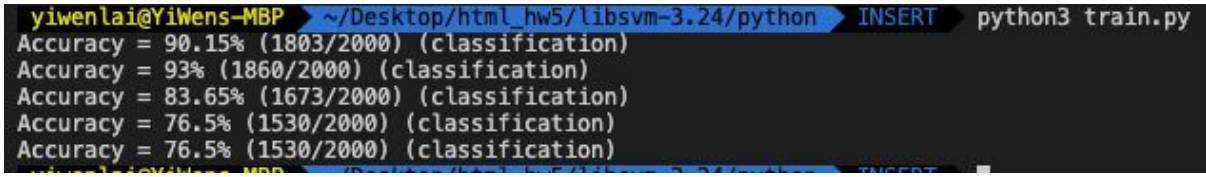
```
# Q 16 17
for classValue in [1,2,3,4,5]:
    y, x = svm_read_problem('hw5_train')

    for i in range(len(y)):
        if y[i] == classValue:
            y[i] = 1
        else:
            y[i] = -1
    m = svm_train(y, x, '-c 10 -s 0 -t 1 -d 2 -r 1 -g
1 -q')
    p_label, p_acc, p_val = svm_predict(y, x, m,'-q')
    support_vectors = m.get_SV()
    ACC, MSE, SCC = evaluations(y, p_label)
    print("Class: ",classValue," Number of support
vectors: ", len(support_vectors)," Accuracy: ",ACC)
```

18. d
    a. result



    b. Code will be provided together with q19

19. b
    a. result



    b. Code for both 18 & 19

```
y, x = svm_read_problem('hw5_train')
y_test,x_test = svm_read_problem('hw5_test')
for i in range(len(y)):
    if y[i] == 6.0:
        y[i] = 1
    else:
```

```
        y[i] = -1
for j in range(len(y_test)):
    if y_test[j]==6.0:
        y_test[j] = 1
    else:
        y_test[j] = -1
for C in [0.01,0.1,1,10,100]: # part for 18
    m = svm_train(y, x, '-c %f -s 0 -t 2 -g 10 -q' %
C)
    p_label, p_acc, p_val = svm_predict(y_test,
x_test, m)
for r in [0.1,1,10,100,1000]: # part for 19
    m = svm_train(y, x, '-c 0.1 -s 0 -t 2 -g %f -q' %
r)
    p_label, p_acc, p_val = svm_predict(y_test,
x_test, m)
```

20. b
    a. result after 1000 iteration

```
999
{0.1: 271, 1: 729, 10: 0, 100: 0, 1000: 0}
yiwenlai@YiWens-MBP ~/Desktop/html_hw5/libsvm-3.24/python INSERT
```

    b. Code

```
y, x = svm_read_problem('hw5_train')

for i in range(len(y)):
    if y[i] == 6.0:
        y[i] = 1
    else:
        y[i] = -1
calculate = {0.1:0,1:0,10:0,100:0,1000:0}
for t in range(1000):
    val_list = []
    while len(val_list)< 200:
        num = random.randint(0,4434)
```

```
        if num not in val_list:
            val_list.append(num)
    x_sub = []
    y_sub = []
    x_val = []
    y_val = []
    for i in range(len(x)):
        if i in val_list:
            x_val.append(x[i])
            y_val.append(y[i])
        else:
            x_sub.append(x[i])
            y_sub.append(y[i])
    max_acc = 0
    rou = 0.1
    for r in [0.1,1,10,100,1000]:
        m = svm_train(y_sub, x_sub, '-c 0.1 -s 0 -t 2
-g %f -q' % r)
        p_label, p_acc, p_val = svm_predict(y_val,
x_val, m,' -q')
        ACC, MSE, SCC = evaluations(y_val, p_label)
        # print(ACC)
        if ACC > max_acc:
            max_acc = ACC
            rou  = r
        elif ACC == max_acc:
            if r < rou:
                rou = r
    calculate[rou] += 1
    print(t)
    print(calculate)
```