Homework#5

B08902071 塗季芸

1. Answer: (d).

$$X = \begin{bmatrix} 1 & -2 & 4 \\ 1 & 0 & 0 \\ 1 & 2 & 4 \end{bmatrix} \quad y = \begin{bmatrix} -1 \\ +1 \\ -1 \end{bmatrix}$$

$$\begin{cases} (1) \quad -w_1 + 2w_2 - 4w_3 - b \geq 1 \\ (2) \quad w_1 + b \geq 1 \\ (3) \quad -w_1 - 2w_2 - 4w_3 - b \geq 1 \end{cases}$$

$$\begin{aligned} (1)\&(2) \quad & w_2 - 2w_3 \geq 1 \\ (2)\&(3) \quad & -w_2 - 2w_3 \geq 1 \end{aligned}$$

Combine the above two results, we have

$$\begin{cases} w_2 \geq 0 \\ w_3 \leq -\frac{1}{2} \end{cases}$$

Therefore

$$\frac{1}{2}\mathbf{w}^T\mathbf{w} \geq \frac{1}{2}(w_1^2 + 0 + \frac{1}{4})$$

Also, w_1 only has the constraint of (2), in which b is an arbitrary real number, so

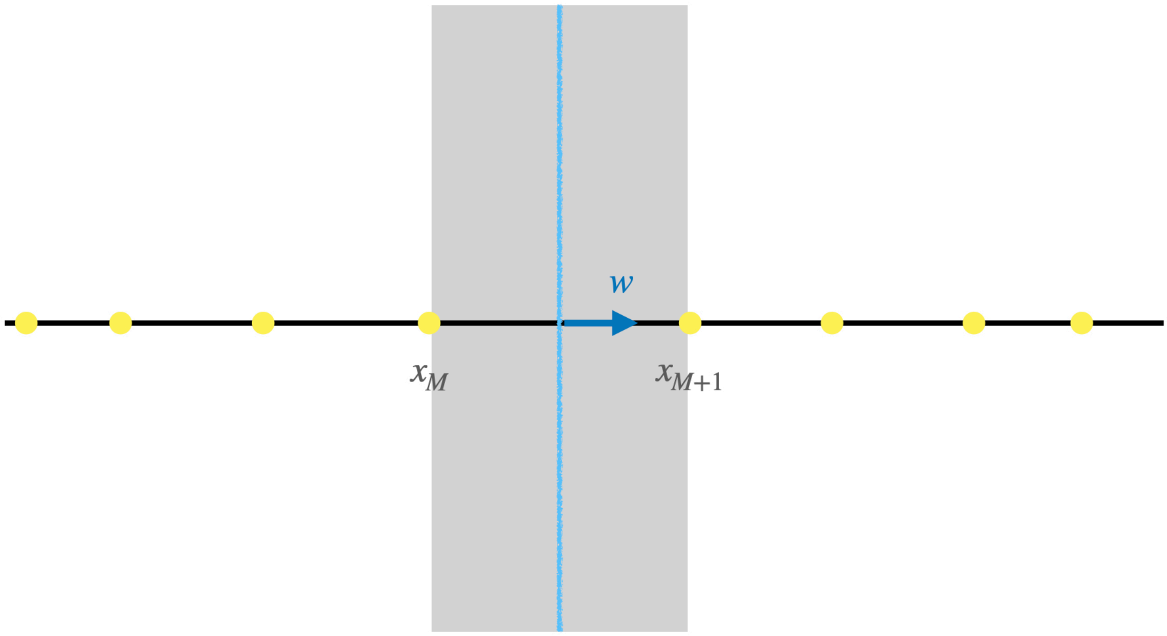
$$w_1^* = 0$$

2. Answer: (b).

$$\text{margin}(b, \mathbf{w}) = \frac{1}{||\mathbf{w}||} = \frac{1}{\sqrt{0 + 0 + \frac{1}{4}}} = 2$$

3. Answer: (e).

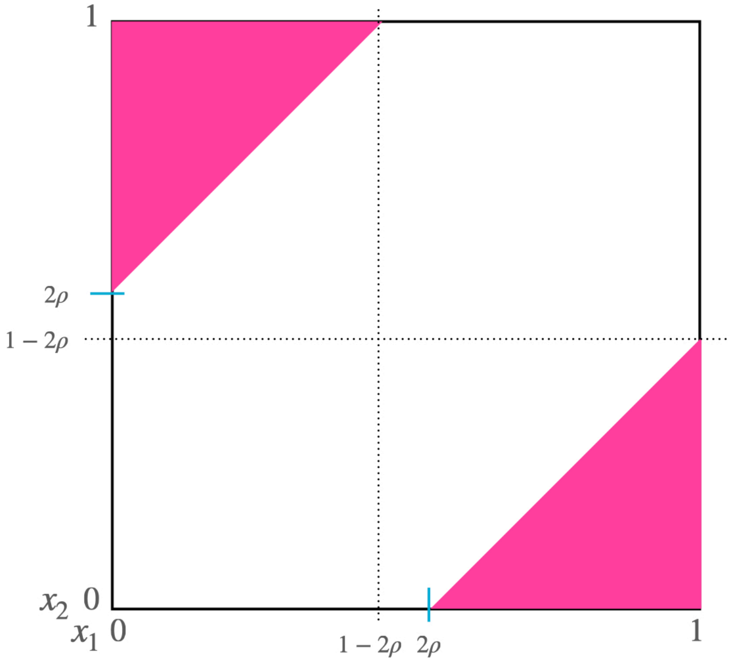
   When $\rho = 0$, the hard-margin SVM is just perceptrons, and we can visualize the optimal perceptron in the figure below:



   The largest margin is obviouly $\frac{1}{2}(x_{M+1} - x_M)$.

4. Answer: (a).

First, consider the probability that $|x_1 - x_2| \geq 2\rho$, which can be illustrated in the figure below.



The probability of sampling x_1, x_2 from a uniform distribution [0,1] such that $|x_1 - x_2| \geq 2\rho$ is $(1-2\rho)^2$.

If $|x_1 - x_2| \geq 2\rho$, the perceptron can be between the two points or outside them, so it can shatter the input. While if $|x_1 - x_2| < 2\rho$, it can only deal with labels of (+,+) and (-,-). The expected number of dichotomies is then $4(1-2\rho)^2 + 2(1-(1-2\rho)^2) = 2 + 2(1-2\rho)^2$.

5. Answer: (c).

   In this problem, we want to solve

$$\max_{all\ \alpha_n \geq 0} \big(\min_{b,\mathbf{w}} \frac{1}{2}\mathbf{w}^T\mathbf{w} + \sum_{n=1}^{N} [\![y_n = +1]\!] \alpha_n(\rho_+ - y_n(\mathbf{w}^T\mathbf{z}_n + b)) + \sum_{n=1}^{N} [\![y_n = -1]\!] \alpha_n \big(\rho_- - y_n(\mathbf{w}^T\mathbf{z}_n + b)\big) \big)$$

   After removing b and $\mathbf{w}$ from the equation using the same steps as p9-p10 in the lecture 202 slides, the problem becomes

$$\max_{all\ \alpha_n \geq 0, \sum y_n\alpha_n = 0, \mathbf{w} = \sum \alpha_n y_n \mathbf{z}_n} -\frac{1}{2}||\sum_{n=1}^{N} \alpha_n y_n \mathbf{z}_n||^2 + \sum_{n=1}^{N} ([\![y_n = +1]\!]\rho_+\alpha_n + [\![y_n = -1]\!]\rho_-\alpha_n)$$

   which is equivalent to solve

$$\min_{all\ \alpha_n \geq 0, \sum y_n\alpha_n = 0, \mathbf{w} = \sum \alpha_n y_n \mathbf{z}_n} \frac{1}{2}||\sum_{n=1}^{N} \alpha_n y_n \mathbf{z}_n||^2 - \sum_{n=1}^{N} ([\![y_n = +1]\!]\rho_+\alpha_n + [\![y_n = -1]\!]\rho_-\alpha_n)$$

6. Answer: (c).

First, since

$$\sum_{n=1}^{N} y_n \alpha_n = 0$$

, therefore,

$$\sum_{n:y_n=1} 1 \cdot \alpha_n = - \sum_{n:y_n=-1} -1 \cdot \alpha_n$$
$$\sum_{n:y_n=1} \alpha_n = \sum_{n:y_n=-1} \alpha_n$$

We can rewrite the last item in the dual problem

$$\begin{aligned}
&- \sum_{n=1}^{N} ([\![y_n = +1]\!]\rho_+ \alpha_n + [\![y_n = -1]\!]\rho_- \alpha_n) \\
&= -\rho_+ \sum_{n:y_n=1} \alpha_n - \rho_- \sum_{n:y_n=-1} \alpha_n \\
&= -(\rho_+ + \rho_-) \sum_{n:y_n=1} \alpha_n \\
&= \frac{-(\rho_+ + \rho_-)}{2} \sum_{n=1}^{N} \alpha_n
\end{aligned}$$

Rewrite the dual problem, we have

$$\begin{aligned}
&\min_{\alpha} \frac{1}{2} || \sum_{n=1}^{N} \alpha_n y_n \mathbf{z}_n ||^2 - \frac{(\rho_+ + \rho_-)}{2} \sum_{n=1}^{N} \alpha_n \\
&= \frac{(\rho_+ + \rho_-)^2}{4} \min_{\alpha} \frac{1}{2} || \sum_{n=1}^{N} \left(\frac{2}{\rho_+ + \rho_-} \alpha_n \right) y_n \mathbf{z}_n ||^2 - \sum_{n=1}^{N} \left(\frac{2}{\rho_+ + \rho_-} \alpha_n \right) \\
&= \frac{(\rho_+ + \rho_-)^2}{4} \left(\frac{1}{2} || \sum_{n=1}^{N} \alpha^* y_n \mathbf{z}_n ||^2 - \sum_{n=1}^{N} \alpha^* \right)
\end{aligned}$$

Let $\alpha^{*\prime}$ be the optimal solution of the uneven-margin SVM,

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

, so

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

7. Answer: (d).

Consider

$$K(\mathbf{x}, \mathbf{x}\prime) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

is symmetric and positive semi-definite.

$$\log_2 K(\mathbf{x}, \mathbf{x}\prime) = \begin{bmatrix} 0 & -\infty \\ -\infty & 0 \end{bmatrix}$$

Let

$$z = \begin{bmatrix} 1 & 1 \end{bmatrix}$$

We have

$$z^T \log_2 K(\mathbf{x}, \mathbf{x}\prime) z = -\infty < 0$$

Therefore, $\log_2 K(\mathbf{x}, \mathbf{x}\prime)$ is not always a valid kernel.

8. Answer: (c).

$$\begin{aligned}
||\phi(\mathbf{x}) - \phi(\mathbf{x}\prime)||^2 &= \phi(\mathbf{x})^T\phi(\mathbf{x}) - 2\phi(\mathbf{x})^T\phi(\mathbf{x}\prime) + \phi(\mathbf{x}\prime)^T\phi(\mathbf{x}\prime) \\
&= K(\mathbf{x}, \mathbf{x}\prime) - 2K(\mathbf{x}, \mathbf{x}\prime) + K(\mathbf{x}\prime, \mathbf{x}\prime) \\
&= \exp(0) - 2\exp(-\gamma||\mathbf{x} - \mathbf{x}\prime||^2) + \exp(0) \\
&\leq 2
\end{aligned}$$

When $||\mathbf{x} - \mathbf{x}\prime|| \to \infty$, $||\phi(\mathbf{x}) - \phi(\mathbf{x}\prime)||^2 = 2$.

9. Answer: (d).

$$\hat{h}(\mathbf{x}_i) = sign\left(\sum_{n\neq i} y_n \exp(-\gamma||\mathbf{x}_n - \mathbf{x}_i||^2) + y_i\right)$$

A worst case can happen when all $y_n, \forall n \neq i$ is different from y_i. To correctly classify such cases, we want

$$\sum_{n\neq i} \exp(-\gamma||\mathbf{x}_n - \mathbf{x}_i||^2) < 1$$

Applying the fact that $||\mathbf{x}_n - \mathbf{x}_i||^2 \geq \epsilon^2$, we have

$$\sum_{n\neq i} \exp(-\gamma||\mathbf{x}_n - \mathbf{x}_i||^2) \leq \sum_{n\neq i} \exp(-\gamma\epsilon^2) = (N-1)\exp(-\gamma\epsilon^2)$$

To ensure that

$$(N-1)\exp(-\gamma\epsilon^2) < 1$$

We have

$$\gamma > \frac{\ln(N-1)}{\epsilon^2}$$

10. Answer: (c).

$$\sum_{n=1}^{N} \alpha_{t,n}\phi(\mathbf{x}_n) + y_{n(t)}\phi(\mathbf{x}_{n(t)}) = \sum_{n \neq n(t)} \alpha_{t,n}\phi(\mathbf{x}_n) + (\alpha_{t,n(t)} + y_{n(t)})\phi(\mathbf{x}_{n(t)})$$

11. Answer: (a).

$$\begin{aligned}
\mathbf{w}_t^T \phi(\mathbf{x}) &= \sum_{n=1}^{N} \alpha_{t,n} \phi(\mathbf{x}_n)^T \phi(\mathbf{x}) \\
&= \sum_{n=1}^{N} \alpha_{t,n} K(\mathbf{x}_n, \mathbf{x})
\end{aligned}$$

12. Answer: (b).

 Because every example is a support vector,

$$b = y_n - y_n\xi_n - \mathbf{w}^T\mathbf{z}_n,\ \forall n$$

 , and $\xi_n \geq 0$. Therefore,

$$\min_{n:y_n<0}\left(-1-\sum_{m=1}^{N} y_m\alpha_m K(x_n,x_m)\right) \leq -1+\xi_n-\sum_{m=1}^{N} y_m\alpha_m K(x_n,x_m)$$

$$1-\xi_n-\sum_{m=1}^{N} y_m\alpha_m K(x_n,x_m) \leq \min_{n:y_n>0}\left(1-\sum_{m=1}^{N} y_m\alpha_m K(x_n,x_m)\right)$$

$$\min_{n:y_n<0}\left(-1-\sum_{m=1}^{N} y_m\alpha_m K(x_n,x_m)\right) \leq b* \leq \min_{n:y_n>0}\left(1-\sum_{m=1}^{N} y_m\alpha_m K(x_n,x_m)\right)$$

13. Answer: (e).

The Lagrange function with Lagrange multipliers α_n is

$$\mathcal{L}(b, \mathbf{w}, \xi, \alpha) = \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{n=1}^{N}\xi_n^2 + \sum_{n=1}^{N}\alpha_n(1 - \xi_n - y_n(\mathbf{w}^T\phi(\mathbf{x}_n) + b))$$

To solve

$$\max_{\alpha_n \geq 0}\left(\min \mathcal{L}(b, \mathbf{w}, \xi, \alpha)\right) \tag{1}$$

we have

$$\frac{\partial \mathcal{L}}{\partial \xi_n} = 0 = 2C\xi_n - \alpha_n$$

therefore,

$$\mathcal{L}(b, \mathbf{w}, \xi, \alpha) = \frac{1}{2}\mathbf{w}^T\mathbf{w} - \frac{1}{4C}\sum_{n=1}^{N}\alpha_n^2 + \sum_{n=1}^{N}\alpha_n(1 - y_n(\mathbf{w}^T\phi(\mathbf{x}_n) + b))$$

Then, apply the same routine that taught in class,

$$\frac{\partial \mathcal{L}}{\partial b} = 0$$
$$\frac{\partial \mathcal{L}}{\partial w_i} = 0$$

Solving (1) is equal to solving

$$\begin{aligned}
&\min_{\alpha} \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N}\alpha_n\alpha_m y_n y_m K(\mathbf{x}_n, \mathbf{x}_m) + \frac{1}{4C}\sum_{n=1}^{N}\alpha_n^2 - \sum_{n=1}^{N}\alpha_n \\
&= \min_{\alpha} \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N}\alpha_n\alpha_m y_n y_m \left(K(\mathbf{x}_n, \mathbf{x}_m) + \frac{1}{2C}[\![n = m]\!]\right) - \sum_{n=1}^{N}\alpha_n
\end{aligned}$$

14. Answer: (e).

    From the condition we already set when solving Q13, we have

$$\frac{\partial \mathcal{L}}{\partial \xi_n} = 0 = 2C\xi_n - \alpha_n$$

which means

$$\xi^* = \frac{\alpha^*}{2C}$$

Experiments with Soft-Margin SVM

Preparation

- build training data according to the classification rule.

```
awk '{if($1==1) $1="+1"; else $1="-1"; print}' satimage.scale > satimage.scale.1
awk '{if($1==2) $1="+1"; else $1="-1"; print}' satimage.scale > satimage.scale.2
awk '{if($1==3) $1="+1"; else $1="-1"; print}' satimage.scale > satimage.scale.3
awk '{if($1==4) $1="+1"; else $1="-1"; print}' satimage.scale > satimage.scale.4
awk '{if($1==5) $1="+1"; else $1="-1"; print}' satimage.scale > satimage.scale.5
awk '{if($1==6) $1="+1"; else $1="-1"; print}' satimage.scale > satimage.scale.6
```

15. Answer: (d).

```
b08902071@linux14 [/tmp2/b08902071/mlt-hw5/libsvm] svm-train -s 0 -t 0 -c 10 satimage.scale.3
........*..*............*
optimization finished, #iter = 22874
nu = 0.108695
obj = -4784.881503, rho = 3.623003
nSV = 500, nBSV = 466
Total nSV = 500
b08902071@linux14 [/tmp2/b08902071/mlt-hw5/libsvm]
```

```cpp
#include <bits/stdc++.h>
using namespace std;
double sum[40];

int main()
{
  ios_base::sync_with_stdio(false); cin.tie(0);
  freopen("satimage.scale.3.model","r",stdin);
  string s;
  for(int i=0;i<8;i++) getline(cin,s);
  while(getline(cin,s)){
    string t;
    stringstream ss(s);
    ss>>t;
    sum[0]=stod(t);
    while(ss>>t){
      istringstream ist(t);
      string tmp;
      int id;
      double x;
      getline(ist,tmp,':');
      id=stoi(tmp);
      getline(ist,tmp);
      x=stod(tmp);
      sum[id]+=sum[0]*x;
    }
  }
  double ans=0.0;
  for(int i=1;i<40;i++) ans+=pow(sum[i],2);
  cout << sqrt(ans) << "\n";
}
```

16. Answer: (b).

```
b08902071@linux14 [/tmp2/b08902071/mlt-hw5/libsvm] svm-train -s 0 -t 1 -d 2 -g 1 -r 1 -c 2 satimage.scale.1
......*....*
optimization finished, #iter = 10837
nu = 0.019633
obj = -115.598741, rho = 0.328482
nSV = 154, nBSV = 50
Total nSV = 154
b08902071@linux14 [/tmp2/b08902071/mlt-hw5/libsvm] svm-predict satimage.scale.1 satimage.scale.1.model
Usage: svm-predict [options] test_file model_file output_file
options:
-b probability_estimates: whether to predict probability estimates, 0 or 1 (default 0); for one-class SVM only 0 is supported
-q : quiet mode (no outputs)
b08902071@linux14 [/tmp2/b08902071/mlt-hw5/libsvm] svm-predict satimage.scale.1 satimage.scale.1.model satimage.scale.1.out
Accuracy = 99.7971% (4426/4435) (classification)
b08902071@linux14 [/tmp2/b08902071/mlt-hw5/libsvm] svm-train -s 0 -t 1 -d 2 -g 1 -r 1 -c 2 satimage.scale.2
.*.*
optimization finished, #iter = 2429
nu = 0.006092
obj = -29.836568, rho = 2.452079
nSV = 87, nBSV = 5
Total nSV = 87
b08902071@linux14 [/tmp2/b08902071/mlt-hw5/libsvm] svm-predict satimage.scale.2 satimage.scale.2.model satimage.scale.2.out
Accuracy = 100% (4435/4435) (classification)
b08902071@linux14 [/tmp2/b08902071/mlt-hw5/libsvm] svm-train -s 0 -t 1 -d 2 -g 1 -r 1 -c 2 satimage.scale.3
.......*....*
optimization finished, #iter = 11233
nu = 0.082421
obj = -673.916212, rho = 3.360598
nSV = 437, nBSV = 306
Total nSV = 437
b08902071@linux14 [/tmp2/b08902071/mlt-hw5/libsvm] svm-predict satimage.scale.3 satimage.scale.3.model satimage.scale.3.out
Accuracy = 97.4295% (4321/4435) (classification)
b08902071@linux14 [/tmp2/b08902071/mlt-hw5/libsvm] svm-train -s 0 -t 1 -d 2 -g 1 -r 1 -c 2 satimage.scale.4
.......*...*
optimization finished, #iter = 10078
nu = 0.154027
obj = -1248.534776, rho = 1.949456
nSV = 770, nBSV = 624
Total nSV = 770
b08902071@linux14 [/tmp2/b08902071/mlt-hw5/libsvm] svm-predict satimage.scale.4 satimage.scale.4.model satimage.scale.4.out
Accuracy = 95.31% (4227/4435) (classification)
b08902071@linux14 [/tmp2/b08902071/mlt-hw5/libsvm] svm-train -s 0 -t 1 -d 2 -g 1 -r 1 -c 2 -q satimage.scale.5
b08902071@linux14 [/tmp2/b08902071/mlt-hw5/libsvm] svm-predict satimage.scale.5 satimage.scale.5.model satimage.scale.5.out
Accuracy = 98.7824% (4381/4435) (classification)
b08902071@linux14 [/tmp2/b08902071/mlt-hw5/libsvm] svm-train -s 0 -t 1 -d 2 -g 1 -r 1 -c 2 -q satimage.scale.6
b08902071@linux14 [/tmp2/b08902071/mlt-hw5/libsvm] svm-predict satimage.scale.6 satimage.scale.6.model satimage.scale.6.out
Accuracy = 95.4228% (4232/4435) (classification)
```

17. Answer: (c).

```
b08902071@linux14 [/tmp2/b08902071/mlt-hw5/libsvm] svm-train -s 0 -t 1 -d 2 -g 1 -r 1 -c 10 satimage.scale.1
............*...*
optimization finished, #iter = 15704
nu = 0.008385
obj = -220.264016, rho = 0.006089
nSV = 145, nBSV = 10
Total nSV = 145
b08902071@linux14 [/tmp2/b08902071/mlt-hw5/libsvm] svm-train -s 0 -t 1 -d 2 -g 1 -r 1 -c 10 satimage.scale.2
.*.*
optimization finished, #iter = 2908
nu = 0.001463
obj = -32.447042, rho = 2.547625
nSV = 87, nBSV = 0
Total nSV = 87
b08902071@linux14 [/tmp2/b08902071/mlt-hw5/libsvm] svm-train -s 0 -t 1 -d 2 -g 1 -r 1 -c 10 satimage.scale.3
...................................*..........................*............*
optimization finished, #iter = 69995
nu = 0.072509
obj = -2872.825882, rho = 6.225147
nSV = 433, nBSV = 244
Total nSV = 433
b08902071@linux14 [/tmp2/b08902071/mlt-hw5/libsvm] svm-train -s 0 -t 1 -d 2 -g 1 -r 1 -c 10 satimage.scale.4
...........................................*.......................*..........*
optimization finished, #iter = 73743
nu = 0.132555
obj = -5221.665416, rho = 2.338587
nSV = 712, nBSV = 499
Total nSV = 712
b08902071@linux14 [/tmp2/b08902071/mlt-hw5/libsvm] svm-train -s 0 -t 1 -d 2 -g 1 -r 1 -c 10 satimage.scale.5
............................*..............*
optimization finished, #iter = 41434
nu = 0.031060
obj = -1009.282166, rho = -2.036284
nSV = 259, nBSV = 68
Total nSV = 259
b08902071@linux14 [/tmp2/b08902071/mlt-hw5/libsvm]
```

18. Answer: (d)(e).

```
b08902071@linux14 [/tmp2/b08902071/mlt-hw5/libsvm] svm-train -s 0 -t 2 -d 2 -g 10 -c 0.01 -q satimage.scale.6
b08902071@linux14 [/tmp2/b08902071/mlt-hw5/libsvm] awk '{if($1==6) $1="+1"; else $1="-1"; print}' satimage.scale.t > satimage.scale.t.6
b08902071@linux14 [/tmp2/b08902071/mlt-hw5/libsvm] svm-predict satimage.scale.t.6 satimage.scale.6.model satimage.scale.t.6.out
Accuracy = 76.5% (1530/2000) (classification)
b08902071@linux14 [/tmp2/b08902071/mlt-hw5/libsvm] svm-train -s 0 -t 2 -d 2 -g 10 -c 0.1 -q satimage.scale.6
b08902071@linux14 [/tmp2/b08902071/mlt-hw5/libsvm] svm-predict satimage.scale.t.6 satimage.scale.6.model satimage.scale.t.6.out
Accuracy = 83.65% (1673/2000) (classification)
b08902071@linux14 [/tmp2/b08902071/mlt-hw5/libsvm] svm-train -s 0 -t 2 -d 2 -g 10 -c 1 -q satimage.scale.6
b08902071@linux14 [/tmp2/b08902071/mlt-hw5/libsvm] svm-predict satimage.scale.t.6 satimage.scale.6.model satimage.scale.t.6.out
Accuracy = 89.35% (1787/2000) (classification)
b08902071@linux14 [/tmp2/b08902071/mlt-hw5/libsvm] svm-train -s 0 -t 2 -d 2 -g 10 -c 10 -q satimage.scale.6
b08902071@linux14 [/tmp2/b08902071/mlt-hw5/libsvm] svm-predict satimage.scale.t.6 satimage.scale.6.model satimage.scale.t.6.out
Accuracy = 90.3% (1806/2000) (classification)
b08902071@linux14 [/tmp2/b08902071/mlt-hw5/libsvm] svm-train -s 0 -t 2 -d 2 -g 10 -c 100 -q satimage.scale.6
b08902071@linux14 [/tmp2/b08902071/mlt-hw5/libsvm] svm-predict satimage.scale.t.6 satimage.scale.6.model satimage.scale.t.6.out
Accuracy = 90.3% (1806/2000) (classification)
b08902071@linux14 [/tmp2/b08902071/mlt-hw5/libsvm]
```

19. Answer: (b).

```
b08902071@linux14 [/tmp2/b08902071/mlt-hw5/libsvm] svm-train -s 0 -t 2 -d 2 -g 0.1 -c 0.1 -q satimage.scale.6
b08902071@linux14 [/tmp2/b08902071/mlt-hw5/libsvm] svm-predict satimage.scale.t.6 satimage.scale.6.model satimage.scale.t.6.out
Accuracy = 90.15% (1803/2000) (classification)
b08902071@linux14 [/tmp2/b08902071/mlt-hw5/libsvm] svm-train -s 0 -t 2 -d 2 -g 1 -c 0.1 -q satimage.scale.6
b08902071@linux14 [/tmp2/b08902071/mlt-hw5/libsvm] svm-predict satimage.scale.t.6 satimage.scale.6.model satimage.scale.t.6.out
Accuracy = 93% (1860/2000) (classification)
b08902071@linux14 [/tmp2/b08902071/mlt-hw5/libsvm] svm-train -s 0 -t 2 -d 2 -g 10 -c 0.1 -q satimage.scale.6
b08902071@linux14 [/tmp2/b08902071/mlt-hw5/libsvm] svm-predict satimage.scale.t.6 satimage.scale.6.model satimage.scale.t.6.out
Accuracy = 83.65% (1673/2000) (classification)
b08902071@linux14 [/tmp2/b08902071/mlt-hw5/libsvm] svm-train -s 0 -t 2 -d 2 -g 100 -c 0.1 -q satimage.scale.6
b08902071@linux14 [/tmp2/b08902071/mlt-hw5/libsvm] svm-predict satimage.scale.t.6 satimage.scale.6.model satimage.scale.t.6.out
Accuracy = 76.5% (1530/2000) (classification)
b08902071@linux14 [/tmp2/b08902071/mlt-hw5/libsvm] svm-train -s 0 -t 2 -d 2 -g 1000 -c 0.1 -q satimage.scale.6
b08902071@linux14 [/tmp2/b08902071/mlt-hw5/libsvm] svm-predict satimage.scale.t.6 satimage.scale.6.model satimage.scale.t.6.out
Accuracy = 76.5% (1530/2000) (classification)
b08902071@linux14 [/tmp2/b08902071/mlt-hw5/libsvm]
```

20. Answer: (b).

```python
from svmutil import *
import random

y, x = svm_read_problem('../satimage.scale.6')
L = len(y)
numlist = [0]*L
for i in range(L):
    numlist[i]=i
y_val, x_val = [0]*200, [0]*200
y_train, x_train = [0]*(L-200), [0]*(L-200)
compare = [0]*5
for rounds in range(1000):
    print(f'round={rounds}, max_index={compare.index(max(compare))}')
    ACC = [0]*5
    cnt_val=0
    cnt_train=0
    S=random.sample(numlist,200)
    for i in range(L):
        #print(f'{cnt_val}, {cnt_train}')
        if(i in S):
            y_val[cnt_val]=y[i]
            x_val[cnt_val]=x[i]
            cnt_val+=1
        else:
            y_train[cnt_train]=y[i]
            x_train[cnt_train]=x[i]
            cnt_train+=1
    m = svm_train(y_train, x_train, '-s 0 -t 2 -c 0.1 -g 0.1 -q')
    p_label, p_acc, p_val = svm_predict(y_val, x_val, m)
    ACC[0], MSE, SCC = evaluations(y_val, p_label)
    m = svm_train(y_train, x_train, '-s 0 -t 2 -c 0.1 -g 1 -q')
    p_label, p_acc, p_val = svm_predict(y_val, x_val, m)
    ACC[1], MSE, SCC = evaluations(y_val, p_label)
    m = svm_train(y_train, x_train, '-s 0 -t 2 -c 0.1 -g 10 -q')
    p_label, p_acc, p_val = svm_predict(y_val, x_val, m)
    ACC[2], MSE, SCC = evaluations(y_val, p_label)
    m = svm_train(y_train, x_train, '-s 0 -t 2 -c 0.1 -g 100 -q')
    p_label, p_acc, p_val = svm_predict(y_val, x_val, m)
    ACC[3], MSE, SCC = evaluations(y_val, p_label)
    m = svm_train(y_train, x_train, '-s 0 -t 2 -c 0.1 -g 1000 -q')
    p_label, p_acc, p_val = svm_predict(y_val, x_val, m)
    ACC[4], MSE, SCC = evaluations(y_val, p_label)
    #compare ACC
    compare[ACC.index(max(ACC))]+=1

print(compare.index(max(compare)))
```