1. (d)

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

$$\begin{cases} -(w_1 - 2w_2 + 4w_3 + b) \geq 1 \\ w_1 + b \geq 1 \\ -(w_1 + 2w_2 + 4w_3 + b) \geq 1 \end{cases} \Rightarrow \begin{cases} -2w_2 - 4w_3 \geq 2 \\ -2w_2 - 4w_3 \geq 2 \Rightarrow w_2 = 0 \\ 4w_2 \geq 0 \end{cases}$$

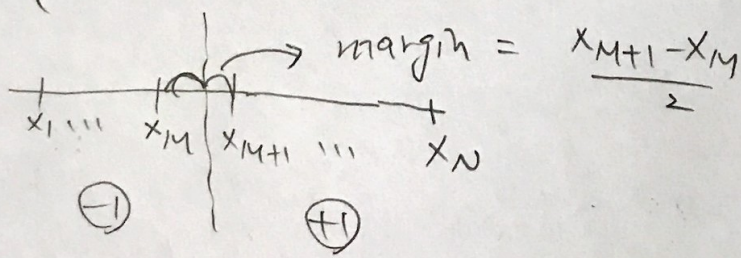
$$\Rightarrow 4w_3 \leq -2 \Rightarrow w_3 = -\frac{1}{2}$$

$$\Rightarrow w_1^* = 0 \quad (\min w^T w)$$

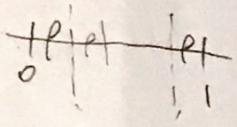
2. (b)

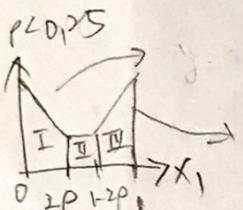
$$\text{margin} = \frac{1}{\|w\|} = \frac{1}{\|(0, 0, -\frac{1}{2})\|} = 2$$

3. (e)



4. (a)





① $(1-2\rho)\cdot 4 + 2\rho\cdot 2$

$= 4-4\rho$

② $(1-4\rho)\cdot 4 + 4\rho\cdot 2$

$= 4-8\rho$

① $\rho < 0.25$

$\frac{(8-12\rho)(2\rho)}{2}$ (I) $+ (4-8\rho)(1-4\rho)$ (II) $+ \frac{(8-12\rho)(2\rho)}{2}$ (III)

$= 16\rho - 24\rho^2 + 4 - 24\rho + 32\rho^2$

$= 8\rho^2 - 8\rho + 4$

$= 2 + (2\rho-1)^2$

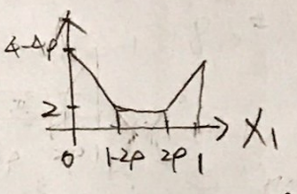
② $0.25 < \rho < 0.5$

$\frac{(6-4\rho)(1-2\rho)}{2}\times 2 + 2(4\rho-1)$

$= 6 - 16\rho + 8\rho^2 + 8\rho - 2$

$= 8\rho^2 - 8\rho + 4$

$= 2 + (2\rho-1)^2$



$(1-2\rho)\cdot 4 + 2\rho\cdot 2$

$= 4-4\rho$

5.

$$L(b, w, \alpha) = \frac{1}{2} w^T w + \sum_{y_n=+1} \alpha_n \left(\rho_+ - y_n(w^T x_n + b)\right) + \sum_{y_n=-1} \alpha_n (\rho_- - y_n(w^T x_n + b))$$

$$\max_{\text{all}\,\alpha_n \geq 0} \left[\min_{b,w} L(b, w, \alpha) \right]$$

① $\frac{\partial L}{\partial b} = 0 \Rightarrow -\sum_{n=1}^{N} \alpha_n y_n = 0 \Rightarrow \max_{\text{all}\,\alpha_n \geq 0, \sum y_n \alpha_n = 0} \left(\min_w \left[\frac{1}{2} w^T w + \sum \alpha_n (\rho_+ - y_n w^T x_n) + \sum \alpha_n (\rho_- - y_n w^T x_n) \right] \right)$

② $\frac{\partial L}{\partial w_i} = 0 \Rightarrow w_i - \sum \alpha_n y_n x_{n,i} = 0$

$\Rightarrow \max_{\text{all}\,\alpha_n \geq 0, -\sum y_n \alpha_n = 0, w = \sum \alpha_n y_n x_n} \left(-\frac{1}{2} \| \sum_{n=1}^{N} \alpha_n y_n x_n \|^2 + \sum_{y_n=+1} \alpha_n \rho_+ + \sum_{y_n=-1} \alpha_n \rho_- \right)$

$\Rightarrow \min \left(\frac{1}{2} \sum_n \sum_m \alpha_n \alpha_m y_n y_m x_n^T x_m - \sum_{y_n=+1} \alpha_n \rho_+ - \sum_{y_n=-1} \alpha_n \rho_- \right)$

6, (e)

SV of uneven margin is the same as SV of even margin

let $w' = kw$

① for $y_n = 1$

$\Rightarrow y_n(w^T x_n + b) \geq 1$

$$\begin{cases} w^T x_n + b \geq 1 & -① \\ w'^T x_n + b \geq \rho^+ \\ kw^T x_n + b \geq \rho^+ & -② \end{cases}$$

② $y_m = -1$

$y_m(w^T x_m + b) \geq 1$

$$\begin{cases} w^T x_m + b \leq -1 & -③ \\ w'^T x_m + b \leq \rho^- \\ kw^T x_m + b \leq \rho^- & -④ \end{cases}$$

① - ② $\Rightarrow w^T(x_n - x_m) \geq 2$

② - ④ $\Rightarrow k w^T (x_n - x_m) \geq \rho^+ - \rho^- \Rightarrow k = \frac{\rho^+ - \rho^-}{2}$

$$w^T(x_n - x_m) \geq \frac{\rho^+ - \rho^-}{k}$$

$w^* = \sum \alpha_n^* y_n z_n$

$w'^* = k w^* = k \sum \alpha_n^* y_n z_n = \sum (k\alpha_n^*) y_n z_n$

$\Rightarrow \alpha'^* = k\alpha^* = \frac{\rho^+ - \rho^-}{2} \alpha^*$

7. (d)

let $K(x_1,x_1)=K(x_2,x_2)=\frac{1}{2}$

$K(x_1,x_2)=K(x_2,x_1)=0$ $\Rightarrow$ matrix $K=\begin{bmatrix}\frac{1}{2} & 0\\ 0 & \frac{1}{2}\end{bmatrix}$ is positive semi-definite

(d) $k'=\log_2 K(x_1,x_2)$

$\Rightarrow K=\begin{bmatrix}-1 & -\infty\\ -\infty & -1\end{bmatrix}$

let $V=\begin{bmatrix}a\\ b\end{bmatrix}$ with $a,b\in R$, $a,b\neq 0$

$V^T K V = -a^2-b^2-\infty\cdot a-\infty\cdot b$

<0 if, $a,b<0$ $\Rightarrow$ not positive semi-definite

$\Rightarrow$ not valid kernels

8. (c)

$\|\phi(x)-\phi(x')\|^2 = \phi(x)^2+\phi(x')^2-2\phi(x)\phi(x')$

$= K(x,x)+K(x',x')-2K(x,x')$

$= 1+1-2\exp(-\gamma\|x-x'\|^2)$

$\leq 2-0=2$

9. (d)

$\hat{h}(x)=\text{sign}\left(\sum_{n=1}^{N} y_n K(x_n,x)\right)$

if $x=x_1$, to let $E_{in}(\hat{h})=0$ $\Rightarrow$ $\hat{h}(x_1)=\text{sign}\left(\sum_{n=1}^{N} y_n e^{-\gamma\|x_n-x_1\|^2}\right)=y_1$

$\Rightarrow \hat{h}(x_1)=\text{sign}\left(y_1+y_2 e^{-\gamma\|x_2-x_1\|^2}+y_3 e^{-\gamma\|x_3-x_1\|^2}+\cdots\right)$

$y_i=\pm 1$ $\Rightarrow$ to let $\hat{h}(x_1)=y_1$ $\Rightarrow$ $\left|\sum_{n=2}^{N} y_n e^{-\gamma\|x_n-x_1\|^2}\right|<1 = y_1$

In worst case $= y_i=1$, $i=2\sim N$, $\|x_n-x_1\|=\varepsilon$

$\Rightarrow \left|\sum_{n=2}^{N} e^{-\gamma\|x_n-x_1\|^2}\right|<1 \Rightarrow (N-1)e^{-\gamma\varepsilon^2}<1$

$\Rightarrow e^{-\gamma\varepsilon^2}<\frac{1}{N-1}$ $\Rightarrow -\gamma\varepsilon^2>\log\frac{1}{N-1}$ $\Rightarrow \gamma\varepsilon^2\geq\log(N-1)$

$\Rightarrow \gamma>\frac{1}{\varepsilon^2}\log(N-1)$, and the same for $x=x_2,x_3,\cdots,x_n$

10. (c)

$$w_{t+1} \leftarrow w_t + y_{n(t)}\phi(x_{n(t)})$$

$$\Rightarrow \sum_{n=1}^{N} \alpha_{t+1,n}\phi(x_n) \leftarrow \sum_{n=1}^{N} \alpha_{t,n}\phi(x_n) + y_{n(t)}\phi(x_{n(t)})$$

if $k = n(t) \Rightarrow \alpha_{t+1,k}\phi(x_k) \leftarrow \alpha_{t,k}\phi(x_k) + y_k\phi(x_k)$

$$\Rightarrow \alpha_{t+1,k} \leftarrow \alpha_{t,k} + y_k$$

$$\Rightarrow \alpha_{t+1,n(t)} \leftarrow \alpha_{t,n(t)} + y_{n(t)}$$

if $k \neq n(t) \Rightarrow \alpha_{t+1,k} \leftarrow \alpha_{t,k}$

$$\Rightarrow \alpha_{t+1,n(t)} \leftarrow \alpha_{t,n(t)}$$

11. (a)

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

$$= \sum_{n=1}^{N}\left(\alpha_{t,n}\phi(x_n)\phi(x)\right)$$

$$= \sum_{n=1}^{N} \alpha_{t,n}K(x_n,x)$$

12. (b)

every sample is bounded SV $\Rightarrow \xi_n > 0 \Rightarrow \beta_n = 0$

$$\Rightarrow \max_{\alpha\geq 0}\left(\min_{b} \frac{1}{2}w^Tw + \sum_n C(1-y_n(w^Tz_n+b))\right)$$

$$b^* = y_n - y_n\xi_n - \sum_m \alpha_m y_m K(x_m, x_n)$$

if $y_n = 1 \Rightarrow b^* = 1 - \xi_n - \sum_m \alpha_m y_m k(x_m, x_n) \leq 1 - \sum \alpha_m y_m k(x_m, x_n)$

$y_n = -1 \Rightarrow b^* = -1 + \xi_n - \sum_m \alpha_m y_m k(x_m, x_n) \geq -1 - \sum_m \alpha_m y_m k(x_m, x_n)$

$$b^* \leq \min_{n;y_n=1}\left(1 - \sum_m \alpha_m y_m k(x_m, x_n)\right)$$

(b^* must be bounded by all the $y_n=1$ samples' upper bound $\Rightarrow$ largest b^* is the min upper bound among all $y_n=1$ samples)

13, (e)

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

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

$$\frac{\partial L}{\partial b} = 0 \Rightarrow -\sum \alpha_n y_n = 0$$

$$\frac{\partial L}{\partial w_i} = 0 \Rightarrow w_i - \sum \alpha_n y_n \phi(x_n)_i = 0 \Rightarrow w = \sum_{n=1}^{N} \alpha_n y_n z_n$$

$$\Rightarrow \max_{\substack{\alpha_n \geq 0 \\ \alpha_n = 2C\xi_n \\ \sum \alpha_n y_n = 0 \\ w = \sum \alpha_n y_n z_n}} \left(-\frac{1}{2} w^T w + C \sum \xi_n^2 - \sum \alpha_n \xi_n + \sum \alpha_n \right)$$

(ξ_n^2 → $2C\xi_n$)

$$= \max \left(-\frac{1}{2} \sum \sum \alpha_n \alpha_m y_n y_m K(x_n, x_m) - C \sum \xi_n^2 + \sum \alpha_n \right)$$

($\xi_n = \frac{\alpha_n}{2C}$)

$$= \min_{\substack{\alpha_n \geq 0 \\ \sum \alpha_n y_n = 0}} \left(\frac{1}{2} \sum_n \sum_m \alpha_n \alpha_m y_n y_m \underbrace{K(x_n, x_m)}_{+} + \frac{1}{4C} \sum \alpha_n^2 - \sum \alpha_n \right)$$

$$= \min_{\substack{\alpha_n \geq 0 \\ \sum \alpha_n y_n = 0}} \left(\frac{1}{2} \sum_n \sum_m \alpha_n \alpha_m y_n y_m \left\{ K(x_n, x_m) + \frac{1}{2C} [\![n = m]\!] \right\} - \sum \alpha_n \right)$$

14, $\xi_n = \frac{\alpha_n}{2C} \Rightarrow$ (e)

```python
from svmutil import *
import random

train_y, train_x = svm_read_problem('../../hw5_train.txt')
test_y, test_x = svm_read_problem('../../hw5_test.txt')
x_dim = 36

# 15
y = [1 if y == 3 else -1 for y in train_y]
model = svm_train(y, train_x, '-c 10 -s 0 -t 0 -q')
w = [0] * x_dim
for i in range(model.get_nr_sv()):
    alpha_y = model.get_sv_coef()[i][0]
    xn = model.get_SV()[i]
    for f_ind in range(x_dim):
        w[f_ind] += alpha_y * xn.setdefault(f_ind+1, 0)
w_norm = 0
for i in range(x_dim):
    w_norm += (w[i] ** 2)
print(w_norm ** (1/2))

# 16, 17
for i in range(1, 6):
    y = [1 if y == i else -1 for y in train_y]
    model = svm_train(y, train_x, '-c 10 -s 0 -t 1 -g 1 -r 1 -d 2 -q')
    _, p_acc, _ = svm_predict(y, train_x, model)
    print(i, p_acc[0], len(model.get_SV()))

# 18
C = [0.01, 0.1, 1, 10, 100]
y = [1 if y == 6 else -1 for y in train_y]
t_y = [1 if y == 6 else -1 for y in test_y]
for c in C:
    model = svm_train(y, train_x, f'-c {c} -s 0 -t 2 -g 10 -q')
    _, p_acc, _ = svm_predict(t_y, test_x, model)
    print(c, p_acc[0])
```

```python
# 19
gamma = [0.1, 1, 10, 100, 1000]
y = [1 if y == 6 else -1 for y in train_y]
t_y = [1 if y == 6 else -1 for y in test_y]
for g in gamma:
    model = svm_train(y, train_x, f'-c 0.1 -s 0 -t 2 -g {g} -q')
    _, p_acc, _ = svm_predict(t_y, test_x, model)
    print(g, p_acc[0])

# 20
def random_get_list(data_y, data_x, num):
    all_rand = {}
    s = 0
    while s < num:
        rand = random.randint(0, len(data_y)-1)
        if rand not in all_rand:
            all_rand[rand] = True
            s += 1

    ty, tx, vy, vx = [], [], [], []
    for i in range(len(data_y)):
        if i in all_rand:
            vy.append(data_y[i])
            vx.append(data_x[i])
        else:
            ty.append(data_y[i])
            tx.append(data_x[i])
    return ty, tx, vy, vx

gamma = [0.1, 1, 10, 100, 1000]
count_times = {0.1: 0, 1: 0, 10: 0, 100: 0, 1000: 0}
```

```python
# 20
def random_get_list(data_y, data_x, num):
    all_rand = {}
    s = 0
    while s < num:
        rand = random.randint(0, len(data_y)-1)
        if rand not in all_rand:
            all_rand[rand] = True
            s += 1

    ty, tx, vy, vx = [], [], [], []
    for i in range(len(data_y)):
        if i in all_rand:
            vy.append(data_y[i])
            vx.append(data_x[i])
        else:
            ty.append(data_y[i])
            tx.append(data_x[i])
    return ty, tx, vy, vx

gamma = [0.1, 1, 10, 100, 1000]
count_times = {0.1: 0, 1: 0, 10: 0, 100: 0, 1000: 0}
y = [1 if y == 6 else -1 for y in train_y]
t_y = [1 if y == 6 else -1 for y in test_y]
for rd in range(1000):
    t_y, t_x, val_y, val_x = random_get_list(train_y, train_x, 200)
    best_g = 0.1
    best_acc = 0
    for g in gamma:
        model = svm_train(t_y, t_x, f'-c 0.1 -s 0 -t 2 -g {g} -q')
        _, p_acc, _ = svm_predict(val_y, val_x, model)
        if p_acc[0] > best_acc:
            best_g = g
            best_acc = p_acc[0]
    count_times[best_g] += 1

print(count_times)
```