5.

no collaborators

$$\begin{cases} \underline{z_n}^T = [1, \underline{x_n}^T] \\ \underline{\tilde{w}^*}^T = [\tilde{w}_0^*, \underline{w^*}^T] \\ \underline{w^*} = [\tilde{w}_1^*, \tilde{w}_2^*, \dots, \tilde{w}_d^*]^T \end{cases}$$

①

$$(\tilde{P}_1) \min_{\tilde{w}, b, \xi} \frac{1}{2}\underline{\tilde{w}}^T\underline{\tilde{w}} + C\sum_{n=1}^{N}\xi_n$$

subject to $y_n(\underline{\tilde{w}}^T \underline{z_n} + b) \geq 1 - \xi_n$, for $n = 1, 2, \dots, N$

$\xi_n \geq 0$, for $n = 1, 2, \dots, N$

$$\mathcal{L}(b, \underline{\tilde{w}}, \underline{\xi}, \underline{\alpha}, \underline{\beta}) = \frac{1}{2}\underline{\tilde{w}}^T\underline{\tilde{w}} + C\sum_{n=1}^{N}\xi_n + \sum_{n=1}^{N}\alpha_n \cdot (1 - \xi_n - y_n(\underline{\tilde{w}}^T \underline{z_n} + b)) + \sum_{n=1}^{N}\beta_n(-\xi_n)$$

$$\max_{\alpha_n \geq 0, \beta_n \geq 0}\left(\min_{b, \underline{\tilde{w}}, \underline{\xi}} \frac{1}{2}\underline{\tilde{w}}^T\underline{\tilde{w}} + C\sum_{n=1}^{N}\xi_n + \sum_{n=1}^{N}\alpha_n \cdot (1 - \xi_n - y_n(\underline{\tilde{w}}^T \underline{z_n} + b)) + \sum_{n=1}^{N}\beta_n(-\xi_n)\right)$$

$$\frac{\partial \mathcal{L}}{\partial \xi_n} = 0 = C - \alpha_n - \beta_n \Rightarrow \beta_n = C - \alpha_n \text{ and } 0 \leq \alpha_n \leq C$$

$$\max_{0 \leq \alpha_n \leq C, \beta_n = C - \alpha_n}\left(\min_{b, \underline{\tilde{w}}, \underline{\xi}} \frac{1}{2}\underline{\tilde{w}}^T\underline{\tilde{w}} + \sum_{n=1}^{N}\alpha_n \cdot (1 - y_n(\underline{\tilde{w}}^T \underline{z_n} + b))\right)$$

$$\frac{\partial \mathcal{L}}{\partial b} = 0 \Rightarrow \sum_{n=1}^{N}\alpha_n y_n = 0$$

$$\frac{\partial \mathcal{L}}{\partial \tilde{w}_i} = 0 \Rightarrow \underline{\tilde{w}} = \sum_{n=1}^{N}\alpha_n y_n \underline{z_n}$$

$$\Rightarrow \underline{\tilde{w}}^* = \sum_{n=1}^{N}\alpha_n y_n [1, \underline{x_n}^T]^T$$

$$= \left[\sum_{n=1}^{N}\alpha_n y_n, \underline{w^*}^T\right]^T$$

$$= [0, \underline{w^*}^T]^T \Rightarrow \tilde{w}_0^* = 0$$

②

$$(\tilde{D}_1) \min_{\underline{\alpha}} \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N}\alpha_n\alpha_m y_n y_m \underline{z_n}^T \underline{z_m} - \sum_{n=1}^{N}\alpha_n$$

subject to $\sum_{n=1}^{N} y_n\alpha_n = 0$, $0 \leq \alpha_n \leq C$, for $n = 1, 2, \dots, N$

$$\frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N}\alpha_n\alpha_m y_n y_m \underline{z_n}^T \underline{z_m} = \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N}\alpha_n\alpha_m y_n y_m [1, \underline{x_n}^T][1, \underline{x_m}^T]^T$$

$$= \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N}\alpha_n\alpha_m y_n y_m (1 + \underline{x_n}^T \underline{x_m})$$

$$= \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N}\alpha_n\alpha_m y_n y_m + \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N}\alpha_n\alpha_m y_n y_m \underline{x_n}^T \underline{x_m}$$

$$= \frac{1}{2}\sum_{n=1}^{N}\alpha_n y_n \sum_{m=1}^{N}\alpha_m y_m + \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N}\alpha_n\alpha_m y_n y_m \underline{x_n}^T \underline{x_m}$$

$$= \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N}\alpha_n\alpha_m y_n y_m \underline{x_n}^T \underline{x_m}$$

$(\tilde{D}_1)\ \min\limits_{\underline{\alpha}} \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} \alpha_n \alpha_m y_n y_m \underline{x_n}^T \underline{x_m} - \sum_{n=1}^{N} \alpha_n$

subject to $\sum_{n=1}^{N} y_n \alpha_n = 0,\ 0 \le \alpha_n \le C$, for $n = 1, 2, \ldots, N$

$\therefore\ \tilde{D}_1 = D_1$

③ solve unique b^* with free SV $(\underline{z_s}, y_s)$:

$$b^* = y_s - \tilde{w}^{*T} \underline{z_s}$$

$$= y_s - [0, \underline{w}^{*T}][1, \underline{x_s}^T]^T$$

$$= y_s - \underline{w}^{*T} \underline{x_s}$$

$\therefore$ By ① & ② & ③ :

$(b^*, \underline{w}^*, \underline{\alpha}^*)$ is also an optimal solution of the original problem (which contains shorter $\underline{x_n}$) and $\tilde{w}_0^* = 0$. Q.E.D.

6. (P) $\min_{\mathbf{w},b,\boldsymbol{\xi}} \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{n=1}^{N}\xi_n$, Let $\boldsymbol{\Phi}(\mathbf{x}_n) = \mathbf{z}_n$

collaborator: B11901073 林禹融

subject to $y_n(\mathbf{w}^T\mathbf{z}_n + b) \geq 1-\xi_n$, for $n=1,2,\ldots,N$

$\xi_n \geq 0$

$y_0(\mathbf{w}^T\mathbf{z}_0 + b) \geq 1$, $y_0 = -1$ and $y_n = +1$

$$\mathcal{L}(b,\mathbf{w},\boldsymbol{\xi},\boldsymbol{\alpha},\boldsymbol{\beta},\gamma) = \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{n=1}^{N}\xi_n + \sum_{n=1}^{N}\alpha_n(1-\xi_n - y_n(\mathbf{w}^T\mathbf{z}_n+b)) + \sum_{n=1}^{N}\beta_n\cdot(-\xi_n) + \gamma(1-y_0(\mathbf{w}^T\mathbf{z}_0+b))$$

want: Lagrange dual

$$\max_{\alpha_n\geq 0,\beta_n\geq 0,\gamma\geq 0}\left(\min_{\mathbf{w},b,\boldsymbol{\xi}} \mathcal{L}(b,\mathbf{w},\boldsymbol{\xi},\boldsymbol{\alpha},\boldsymbol{\beta},\gamma)\right)$$

$$\Rightarrow \max_{\alpha_n\geq 0,\beta_n\geq 0,\gamma\geq 0}\left(\min_{\mathbf{w},b,\boldsymbol{\xi}} \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{n=1}^{N}\xi_n + \sum_{n=1}^{N}\alpha_n(1-\xi_n - y_n(\mathbf{w}^T\mathbf{z}_n+b)) + \sum_{n=1}^{N}\beta_n\cdot(-\xi_n) + \gamma(1-y_0(\mathbf{w}^T\mathbf{z}_0+b))\right)$$

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

$$\Rightarrow \beta_n = C-\alpha_n \text{ and } 0\leq\alpha_n\leq C$$

$$\Rightarrow \max_{0\leq\alpha_n\leq C,\ \beta_n=C-\alpha_n,\ \gamma\geq 0}\left(\min_{\mathbf{w},b,\boldsymbol{\xi}} \frac{1}{2}\mathbf{w}^T\mathbf{w} + \sum_{n=1}^{N}\alpha_n(1-y_n(\mathbf{w}^T\mathbf{z}_n+b)) + \gamma(1-y_0(\mathbf{w}^T\mathbf{z}_0+b))\right)$$

$$\frac{\partial\mathcal{L}}{\partial b} = 0 = -\sum_{n=1}^{N}\alpha_n y_n + \gamma \Rightarrow \gamma = \sum_{n=1}^{N}\alpha_n y_n = \sum_{n=1}^{N}\alpha_n \geq 0$$

$$\Rightarrow \max_{0\leq\alpha_n\leq C,\ \beta_n=C-\alpha_n,\ \gamma=\sum_{n=1}^{N}\alpha_n\geq 0}\left(\min_{\mathbf{w},b,\boldsymbol{\xi}} \frac{1}{2}\mathbf{w}^T\mathbf{w} + \sum_{n=1}^{N}\alpha_n\left(2-\mathbf{w}^T(\mathbf{z}_n-\mathbf{z}_0)\right)\right)$$

$$\frac{\partial\mathcal{L}}{\partial w_i} = 0 = w_i - \sum_{n=1}^{N}\alpha_n(z_{n,i} - z_{0,i})$$

$$\Rightarrow \mathbf{w} = \sum_{n=1}^{N}\alpha_n(\mathbf{z}_n - \mathbf{z}_0)$$

$$\Rightarrow \max_{0\leq\alpha_n\leq C,\ \beta_n=C-\alpha_n,\ \gamma=\sum_{n=1}^{N}\alpha_n\geq 0}\left(-\frac{1}{2}\mathbf{w}^T\mathbf{w} + 2\sum_{n=1}^{N}\alpha_n\right)$$

$$\min_{\alpha}\left(\frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N}\alpha_n\alpha_m(\underline{z}_n-\underline{z}_0)^T(\underline{z}_m-\underline{z}_0)-2\sum_{n=1}^{N}\alpha_n\right)$$

subject to $\sum_{n=1}^{N}\alpha_n \geq 0$

$0 \leq \alpha_n \leq C$, for $n=1,2,\dots,N$

implicitly $\underline{\omega}=\sum_{n=1}^{N}\alpha_n(\underline{z}_n-\underline{z}_0)$

$\beta_n = C-\alpha_n$, for $n=1,2,\dots,N$

$\gamma=\sum_{n=1}^{N}\alpha_n$

$$\therefore \begin{cases} q_{n,m}=(\underline{z}_n-\underline{z}_0)^T(\underline{z}_m-\underline{z}_0)=\left(\underline{\Phi(x_n)}-\underline{\Phi(x_0)}\right)^T\left(\underline{\Phi(x_m)}-\underline{\Phi(x_0)}\right) \\ \underline{p}=-\underline{2}_N \\ \underline{a}_n^T = n\text{-th unit direction} \Rightarrow A = I_{N\times N} \\ \underline{c}=\underline{0}_N \end{cases}$$

no collaborators

7. $h_{1,0}(\underline{x}) = \text{sign}\left(\sum_{n=1}^{N} y_n K(\underline{x_n}, \underline{x})\right)$

$\Rightarrow \hat{h}(\underline{x}) = \text{sign}\left(\sum_{n=1}^{N} y_n \exp(-\gamma \|\underline{x_n} - \underline{x}\|^2)\right)$

$\|\underline{x_n} - \underline{x_m}\| \geq \epsilon$

For correct classification at $\underline{x_i}$ ($i \in \mathbb{N}$, $1 \leq i \leq n$),

$y_i \hat{h}(\underline{x_i}) > 0$

$\Rightarrow y_i \,\text{sign}\left(\sum_{n=1}^{N} y_n \exp(-\gamma \|\underline{x_n} - \underline{x_i}\|^2)\right) > 0$

$\Rightarrow y_i \left(\sum_{n=1}^{N} y_n \exp(-\gamma \|\underline{x_n} - \underline{x_i}\|^2)\right) > 0$

$\Rightarrow y_i \left(y_i \exp(-\gamma \|\underline{x_i} - \underline{x_i}\|^2) + \sum_{n \neq i} y_n \exp(-\gamma \|\underline{x_n} - \underline{x_i}\|^2)\right) > 0$

$\Rightarrow y_i^2 \times e^0 + y_i \sum_{n \neq i} y_n \exp(-\gamma \|\underline{x_n} - \underline{x_i}\|^2) > 0 \quad - \quad (*)$

$\|\underline{x_n} - \underline{x_i}\| \geq \epsilon$

$\Rightarrow \|\underline{x_n} - \underline{x_i}\|^2 \geq \epsilon^2$

$\Rightarrow -\gamma \|\underline{x_n} - \underline{x_i}\|^2 \leq -\gamma \epsilon^2$

$\Rightarrow \exp(-\gamma \|\underline{x_n} - \underline{x_i}\|^2) \leq \exp(-\gamma \epsilon^2)$

$\Rightarrow \left| \sum_{n \neq j} y_n \exp(-\gamma \|\underline{x_n} - \underline{x_i}\|^2) \right| \leq (N-1) \exp(-\gamma \epsilon^2)$

$\Rightarrow -(N-1) \exp(-\gamma \epsilon^2) \leq \sum_{n \neq j} y_n \exp(-\gamma \|\underline{x_n} - \underline{x_i}\|^2) \leq (N-1) \exp(-\gamma \epsilon^2)$

$(*) \Rightarrow 1 - (N-1) \exp(-\gamma \epsilon^2) > 0$

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

$\Rightarrow \exp(-\gamma \epsilon^2) < \frac{1}{N-1}$

$\Rightarrow -\gamma \epsilon^2 < \ln\left(\frac{1}{N-1}\right)$

$\Rightarrow \gamma > \frac{\ln(N-1)}{\epsilon^2}$ Q.E.D.

no collaborators

8. According to P20 of lecture 203,
the necessary & sufficient conditions for a valid kernel:
Mercer's condition:
$\underline{\underline{K}}$ is symmetric and positive semi-definite ($\because \underline{\underline{K}} = \underline{\underline{Z}}\,\underline{\underline{Z}}^T$)

Let $K_c(x, x') = \cos(x-x')$

$$= \cos x \cos x' + \sin x \sin x'$$
$$= \underline{\phi(x)}^T \underline{\phi(x')}, \quad \underline{\phi(x)} = \begin{bmatrix} \cos x \\ \sin x \end{bmatrix}$$

$\therefore K_c$ is a valid kernel

$$K(x,x') = \exp(2K_c(x,x') - 2)$$
$$= \exp(-2)\exp(\gamma K_c(x,x')), \quad \gamma = 2$$

$\exp(-2)$ is a const and always positive
Similar to p.12 of lecture 203,
the exponential of any valid kernel is also a valid kernel
$\therefore K(x,x') = \exp(2(\cos(x-x')) - 2)$ is a valid kernel.

9. no collaborators

$k = (R-0.5) - (L+0.5) + 1$

$= R - L - 1 + 1$

$= R - L$

$$g_{i,\theta_j}(\underline{x}) \cdot g_{i,\theta_j}(\underline{x}') = \begin{cases} 1, & \min(x_i, x_i') > \theta_j \\ 0, & \min(x_i, x_i') \le \theta_j \end{cases}$$

$$K_{ds}(\underline{x}, \underline{x}') = (\underline{\Phi}_{ds}(\underline{x}))^T (\underline{\Phi}_{ds}(\underline{x}'))$$

$$= \sum_{i=1}^{d} \sum_{j=1}^{k} g_{i,\theta_j}(\underline{x}) \cdot g_{i,\theta_j}(\underline{x}')$$

$$= \sum_{i=1}^{d} \max\{\underbrace{0}_{\text{if } \min(x_i, x_i') \le \theta_j}, \min(\underbrace{\lfloor \min(x_i, x_i') - (L+0.5) \rfloor + 1}_{\text{number of thresholds that the } \min(x_i, x_i') \text{ exceeds}}, \underbrace{R-L}_{\text{maximum number of thresholds}})\}$$

10.

collaborator: B11901073 林禹融

```
Results:
C     Q     #SV
0.1   2     505
0.1   3     547
0.1   4     575
1     2     505
1     3     547
1     4     575
10    2     505
10    3     547
10    4     575

Combination with the smallest number of support vectors: C=0.1, Q=2, #SV=505
```

Q\C	0.1	1	10
2	505	505	505
3	547	547	547
4	575	575	575

According to the table, #SV is positively correlated with Q.

hw6-10.py > ...

```python
import os
import requests
import bz2
from libsvm.svmutil import *

train_url = "https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/multiclass/mnist.scale.bz2"
train_file_compressed = "mnist.scale.bz2"
train_file = "mnist.scale"

# Step 1: Download and decompress the data
def download_and_extract(url, dest_path):...

download_and_extract(train_url, train_file_compressed)

# Step 2: Load and preprocess the data
def load_data(file_path):
    y, x = svm_read_problem(file_path)
    labels, features = [], []
    for i in range(len(y)):
        if y[i] == 3 or y[i] == 7:
            labels.append(1 if y[i] == 3 else -1)  # Map class 3 to +1 and class 7 to -1
            features.append(x[i])
    return labels, features

labels, features = load_data(train_file)

# Step 3: Train SVM and count support vectors
def train_and_count_sv(labels, features, C, Q):
    param = f'-t 1 -d {Q} -c {C} -g 1 -r 1 -q' # -t 1: polynomial kernel
    model = svm_train(labels, features, param)
    return model.get_nr_sv()  # Return number of support vectors

C_values = [0.1, 1, 10]
Q_values = [2, 3, 4]
results = []

# Step 4: Test different combinations of C and Q
for C in C_values:
    for Q in Q_values:
        num_sv = train_and_count_sv(labels, features, C, Q)
        results.append((C, Q, num_sv))
        print(f"C={C}, Q={Q}, Support Vectors={num_sv}")

# Step 5: Output results in a table
print("\nResults:")
print(f"{'C':<5} {'Q':<5} {'#SV':<5}")
for C, Q, num_sv in results:
    print(f"{C:<5} {Q:<5} {num_sv:<5}")
```

11.

collaborator: B11901073 林禹融

```
Results:
C     gamma Margin
0.1   0.1   0.048160
0.1   1     0.044893
0.1   10    0.045122
1     0.1   0.011144
1     1     0.004514
1     10    0.004512
10    0.1   0.010969
10    1     0.004320
10    10    0.004282

Combination with the largest margin: C=0.1, gamma=0.1, Margin=0.048160
```

Gamma \ C	0.1	1	10
0.1	0.048160	0.011144	0.010969
1	0.044893	0.004514	0.004320
10	0.045122	0.004512	0.004282

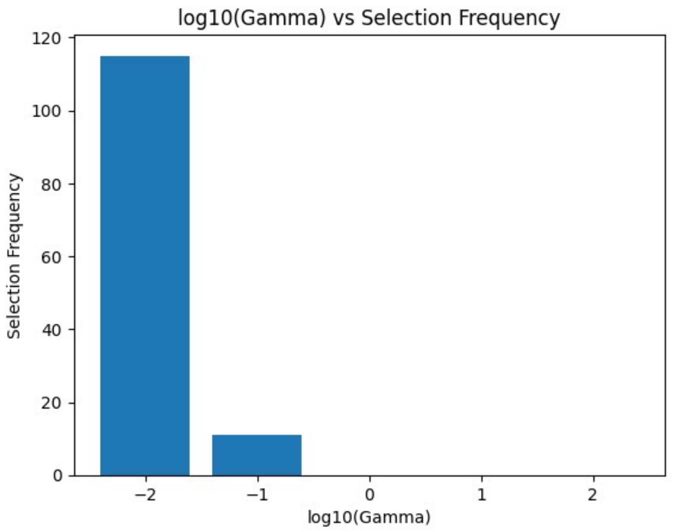
As C increases, the margin generally decreases.
Similarly, as γ increases, the margin decrease.

hw6-11.py > ...

```python
import os…

train_url = "https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/multiclass/mnist.scale.bz2"
train_file_compressed = "mnist.scale.bz2"
train_file = "mnist.scale"

# Step 1: Download and decompress the data
def download_and_extract(url, dest_path):…

download_and_extract(train_url, train_file_compressed)

# Load the data
labels, features = svm_read_problem(train_file)

C_values = [0.1, 1, 10]
gamma_values = [0.1, 1, 10]

# Function to train the model and calculate the margin
def train_and_calculate_margin(labels, features, C, gamma):
    param = f'-t 2 -c {C} -g {gamma} -q'  # Gaussian kernel
    model = svm_train(labels, features, param)
    sv_coef = model.get_sv_coef()
    sv = model.get_SV()

    # Calculate ||w||: Sum the square of sv_coef values
    norm_w_squared = sum([sv_coef[i][0]**2 for i in range(len(sv_coef))])
    norm_w = norm_w_squared**0.5

    margin = 1 / norm_w
    return margin

# Train models and calculate margins
results = []
for C in C_values:
    for gamma in gamma_values:
        margin = train_and_calculate_margin(labels, features, C, gamma)
        results.append((C, gamma, margin))
        print(f"C={C}, gamma={gamma}, Margin={margin}")

# Output results in a table
print("\nResults:")
print(f"{'C':<5} {'gamma':<5} {'Margin':<10}")
for C, gamma, margin in results:
    print(f"{C:<5} {gamma:<5} {margin:<10.6f}")

# Find combination with the largest margin
max_margin = max(results, key=lambda x: x[2])
print(f"\nCombination with the largest margin: C={max_margin[0]}, gamma={max_margin[1]}, Margin={max_margin[2]}")
```

collaborator:
B11901073 林禹融

12.



(x-axis in $\log_{10}$ scale)

According to the bar chart, smaller γ ($= 0.01 \vee 0.1$) are more effective in minimizing the validation error.

hw6-12.py > ...

```python
import os ⋯

train_url = "https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/multiclass/mnist.scale.bz2"
train_file_compressed = "mnist.scale.bz2"
train_file = "mnist.scale"

# Step 1: Download and decompress the data
def download_and_extract(url, dest_path): ⋯

download_and_extract(train_url, train_file_compressed)

def load_data():
    y, x = svm_read_problem(train_file)
    return np.array(y), np.array(x)

# Train and evaluate the SVM model
def train_and_evaluate(y_train, x_train, y_val, x_val, C, gamma):
    model = svm_train(y_train, x_train, f'-t 2 -c {C} -g {gamma} -q')
    _, p_acc, _ = svm_predict(y_val, x_val, model, '-q')
    return 100 - p_acc[0]

y, x = load_data()
C = 1
gamma_values = [0.01, 0.1, 1, 10, 100]
gamma_selection_count = Counter()

for _ in range(128):
    # Split the data into training and validation sets
    x_train, x_val, y_train, y_val = train_test_split(x, y, test_size=200, stratify=y)

    errors = []
    for gamma in gamma_values:
        error = train_and_evaluate(y_train, x_train, y_val, x_val, C, gamma)
        errors.append((gamma, error))

    # Select the gamma with the smallest error, break ties by choosing the smallest gamma
    best_gamma = min(errors, key=lambda item: (item[1], item[0]))[0]
    gamma_selection_count[best_gamma] += 1


# Plot the results
log_gamma_values = [math.log10(gamma) for gamma in gamma_selection_count.keys()]
plt.bar(log_gamma_values, gamma_selection_count.values())
plt.xlabel('log10(Gamma)')
plt.ylabel('Selection Frequency')
plt.title('log10(Gamma) vs Selection Frequency')
plt.show()
```