Machine learning:-

-: Homework 3 :-

① Distance between two hyperplanes, $\{x \in \mathbb{R}^n \mid w^T x = \beta_1\}$ &
$\{x \in \mathbb{R}^n \mid w^T x = \beta_2\}$

→ Let the two hyperplanes be,

$$w^T x_1 = \beta_1 \quad \text{and} \quad w^T x_2 = \beta_2$$

and let's minimize $\|x_1 - x_2\|^2$.

So, the optimization problem is,

$$\underset{x_1, x_2}{\text{minimize}} \ \|x_1 - x_2\|^2$$

$$\text{subject to,} \quad w^T x_1 = \beta_1,$$

$$w^T x_2 = \beta_2$$

combining constraints, $w^T(x_1 - x_2) = \beta_1 - \beta_2$ | constraint ① − constraint ②

By cauchy schwarz inequality, $|a^T b| \le \|a\| \|b\|$

The equation ~~reduce~~ becomes,

$$\|w\| \|x_1 - x_2\| \ge |\beta_1 - \beta_2|$$

$$\therefore \|x_1 - x_2\| \ge \frac{|\beta_1 - \beta_2|}{\|w\|} \longrightarrow ①$$

from, $w^T x_1 = \beta_1$ and $w^T x_2 = \beta_2$ we can write,

$$x_1 = \frac{\beta_1}{\|w\|} w \qquad x_2 = \frac{\beta_2}{\|w\|} w$$

~~Hence~~

$$\|x_1 - x_2\| = \frac{|\beta_1 - \beta_2|}{\|w\|} \longrightarrow ②$$

which is the lowerbound of the inequality ①.

∴ The distance between hyperplanes,

$$\frac{|\beta_1 - \beta_2|}{\|w\|}$$

② $\phi(t) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{t} e^{-x^2/2} dx$

fact:- ϕ is log-concave if $\phi''(t)\,\phi(t) \le (\phi'(t))^2$

(a) Verify $\phi''(t)\,\phi(t) \le (\phi'(t))^2$ for $t \ge 0$

$$\phi'(t) = \frac{1}{\sqrt{2\pi}} e^{-t^2/2}$$

$$\phi''(t) = \frac{-1}{\sqrt{2\pi}} e^{-t^2/2} \cdot \frac{\cancel{2}t}{\cancel{2}}$$

$$\phi''(t) = \frac{-t}{\sqrt{2\pi}} \cdot e^{-t^2/2}$$

LHS, $\phi''(t)\,\phi(t) \Rightarrow -\frac{t}{\sqrt{2\pi}} e^{-t^2/2} \cdot \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{t} e^{-x^2/2} dx$

for $t \ge 0$, LHS is always going to be negative

∴ $\phi''(t)\,\phi(t) \le 0$ | Its $= 0$ when $t = 0$,

RHS, $(\phi'(t))^2 = \left(\frac{1}{\sqrt{2\pi}} e^{-t^2/2}\right)^2 \ge 0$ for $t \ge 0$

∴ $\phi''(t)\,\phi(t) \le (\phi'(t))^2$ for $t \ge 0$

(b) $\frac{x^2}{2} \geq -\frac{t^2}{2} + tx$ verify

→ We know differentiable f with convex domain follows,

$$f(\theta) \geq f(\tilde{\theta}) + \nabla f(\tilde{\theta})^T(\theta - \tilde{\theta})$$

$f(\theta) = t^2/2$, and $t^2/2$ is convex and substituting it

$$\frac{t^2}{2} \geq \frac{x^2}{2} + \frac{\cancel{2}x}{\cancel{2}}(t-x)$$

$$\frac{t^2}{2} \geq -\frac{x^2}{2} + tx$$

$$-\frac{t^2}{2} \leq \frac{x^2}{2} - tx$$

$$\frac{-t^2}{2} + tx \leq \frac{x^2}{2}$$ and it holds with no assumptions on x and t.

(c) From ~~Rearrange~~ (b),

$$\frac{x^2}{2} \geq \frac{-t^2}{2} + tx$$

$$-\frac{x^2}{2} \leq \frac{t^2}{2} - tx$$

Since it holds for all x and t taking exponent wont affect inequality.

$$\therefore e^{-x^2/2} \leq e^{t^2/2 - tx}$$

Integrate over x from $-\infty$ to t

$$\int_{-\infty}^{t} e^{-x^2/2}\,dx \le \int_{-\infty}^{t} e^{t^2/2 - tx}\,dx$$

Simplifying,

$$\int_{-\infty}^{t} e^{-x^2/2}\,dx \le e^{t^2/2}\int_{-\infty}^{t} e^{-tx}\,dx$$

(d) Verify $\phi''(t)\,\phi(t) \le (\phi'(t))^2$ for $t<0$.

$\longrightarrow$ LHS and RHS from (a) and comparing / reducing them

LHS RHS

$$-\frac{t}{\cancel{\sqrt{2\pi}}}\cancel{e^{-t^2/2}} \cdot \frac{1}{\cancel{\sqrt{2\pi}}}\int_{-\infty}^{t} e^{-x^2/2}\,dx \;\boxed{}\; \frac{1}{\cancel{\sqrt{2\pi}}}\cancel{e^{-t^2/2}} \cdot \frac{1}{\cancel{\sqrt{2\pi}}} e^{-t^2/2}$$

$$-t\int_{-\infty}^{t} e^{-x^2/2}\,dx \;\boxed{}\; e^{-t^2/2}$$

We dont know the inequality yet.

$\longrightarrow$ (1)

from (c), $\int_{-\infty}^{t} e^{-x^2/2}\,dx \le e^{t^2/2}\int_{-\infty}^{t} e^{-tx}\,dx$ $\longrightarrow$ (2)

$$\left.\frac{-e^{-tx}}{t}\right|_{-\infty}^{t}$$

$\because \lim_{x\to-\infty} e^{-tx} = 0$ since $t<0$

$\therefore$ $-\frac{e^{-t^2}}{t} + 0$

$\therefore$ $(\rightarrow) -\frac{e^{-t^2}}{t}$ substituting t in ②

$$\int_{-\infty}^{t} e^{x^2/2}\,dx \le e^{t^2/2} \cdot \frac{-e^{-t^2}}{t}$$

$$\int_{-\infty}^{t} e^{-x^2/2}\,dx \le -\frac{e^{-t^2/2}}{t}$$

Since $-t$ is >0 since $t<0$, multiplying both sides by $-t$, doesnt change inequality

$$-t\int_{-\infty}^{t} e^{-x^2/2}\,dx \le e^{-t^2/2}$$

This is LHS $\le$ RHS from ①.

$\therefore$ $\phi''(t)\,\phi(t) \le (\phi'(t))^2$ for $t<0$.

③ $\theta \in \mathbb{R}^m$, $\mathcal{X}$ set. $p(x;\theta) = a(\theta)\exp(\theta^T\phi(x))$

$x \in \mathcal{X}$ $\phi : \mathcal{X} \rightarrow \mathbb{R}^m$

$a(\theta) = \left(\int_{\mathcal{X}} \exp(\theta^T\phi(x))\,dx\right)^{-1}$ $p(x;\theta)$ is density

$a(\theta) = \left(\sum_{x\in\mathcal{X}} \exp(\theta^T\phi(x))\right)^{-1}$ $p(x;\theta)$ is distribution

Show that $\log p(x;\theta)$ is concave in θ.

$\longrightarrow \quad p(x;\theta) = a(\theta)\exp(\theta^T\phi(x))$

$\log p(x;\theta) = \log(a(\theta)) + \log(\exp(\theta^T\phi(x)))$

$\log(\exp(\theta^T\phi(x))) = \theta^T\phi(x)$

is linear in θ, hence nothing can be said about convex / concave of $\log(p(x;\theta))$.

lets take, $\log a(\theta)$ when $\mathcal{X}$ is finite.

$$\log(a(\theta)) = \log\left(\left(\sum_{x\in\mathcal{X}} \exp(\theta^T\phi(x))\right)^{-1}\right)$$

$$= \underbrace{-\log\left(\sum \exp(\theta^T\phi(x))\right)}$$

This expression is of form log-sum-exp which is known to be convex.

$\therefore -\log(\sum \exp(\theta^T\phi(x)))$ is concave.

$\therefore \log(a(\theta))$ is concave

$\therefore \log p(x;\theta)$ is concave when $\forall x \in \mathcal{X}$ and $\mathcal{X}$ is finite.

Since sum of concave and linear function is concave

For discrete but infinite $\mathcal{X}$ and continuous $\mathcal{X}$ can be handled by taking limits of finite sums as the concavity property is preserved under pointwise limits.

The maximum likelihood estimate of $p(x;\theta)$,

$$\max_{\theta} \log p(x;\theta) \Rightarrow \min_{\theta} -\log p(x;\theta)$$

and since $-\log(p(x;\theta))$ is convex MLE for $p(x;\theta)$ is convex optimization problem.

Homework3_4

March 25, 2025

0.0.1 4

[40]:
```
import cvxpy as cp
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import StandardScaler
import pandas as pd
import numpy as np
```

[25]:
```
def mean_absolute_error(y_true, y_pred):
    mae = 0
    for i in range(len(y_true)):
        mae += abs(y_true[i] - y_pred[i])
    return mae/len(y_true)
```

[34]:
```
df = pd.read_csv("wine+quality/winequality-red.csv", sep=';')
X = df.iloc[:, :-1].values
y = df.iloc[:, -1].values
X_train, y_train = X[:1400], y[:1400]
X_test, y_test = X[1400:], y[1400:]

# Standardize the features (x-mean(x))/std(x)
scaler = StandardScaler()
# Use X_train to get the mean and standard deviation
X_train_std = scaler.fit_transform(X_train)
# Use the mean and standard deviation from X_train to standardize X_test
# as having X_train and X_test with same standardization terms makes sense.
X_test_std = scaler.transform(X_test)

def addBiasToSamples(data):
    return np.hstack([data, np.ones((data.shape[0], 1))])

X_train_std = addBiasToSamples(X_train_std)
X_test_std = addBiasToSamples(X_test_std)
```

[52]:
```
# Model 1 using least square loss
model1 = LinearRegression().fit(X_train_std, y_train)
y_pred1 = model1.predict(X_test_std)
print(mean_absolute_error(y_test, y_pred))
```

0.5329671066366284

[54]:
```
# Model 2 using Huber loss
w = cp.Variable(X_train.shape[1])
b = cp.Variable()
r = X_train @ w + b - y_train

obj_fn = cp.sum(cp.huber(r, M=0.5))
model2 = cp.Problem(cp.Minimize(obj_fn))
obj_values = model2.solve()

y_pred2 = X_test @ w.value + b.value
print(mean_absolute_error(y_test, y_pred2))
```

0.5304108520987247

[60]:
```
# Model 3 using hinge loss
w = cp.Variable(X_train.shape[1])
b = cp.Variable()
r = X_train @ w + b - y_train

obj_fn = cp.sum(cp.maximum(0, cp.abs(r) - 1/2 ))
model3 = cp.Problem(cp.Minimize(obj_fn))
obj_values = model3.solve()

y_pred3 = X_test @ w.value + b.value
print(mean_absolute_error(y_test, y_pred3))
```

0.5481057947573722

[ ]:
```
```

Homework3_5

March 25, 2025

0.0.1 5

[26]:
```
import pandas as pd
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
import numpy as np
import cvxpy as cp
```

[27]:
```
df = pd.read_csv(r'/Users/rohitbogulla/Desktop/Sem 2/ML/Assignments/3/
 ↪ionosphere/ionosphere.data', header=None)
X = df.iloc[:, :-1].values
y = df.iloc[:, -1].values

y = (y == 'g').astype(int) * 2 - 1
```

[28]:
```
# The dataset in original form has last 51 samples as positive which is giving␣
 ↪very high
# accuracy for predicting all 1's. Hence randomizing the data split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
 ↪random_state=2)

# Standardizing features
scaler = StandardScaler()
X_train_std = scaler.fit_transform(X_train)
X_test_std = scaler.transform(X_test)

# adding constant term to each sample
X_train_std = np.hstack([np.ones((X_train_std.shape[0], 1)), X_train_std])
X_test_std = np.hstack([np.ones((X_test_std.shape[0], 1)), X_test_std])
```

[29]:
```
# Classifier 1 Least square loss
classifier1 = LinearRegression()
classifier1.fit(X_train, y_train)
y_pred = np.sign(classifier1.predict(X_test))
```

```
print("Accuracy of Classifier using least square loss =",␣
↪accuracy_score(y_test, y_pred))
```

```
Accuracy of Classifier using least square loss = 0.7605633802816901
```

[30]:
```
# Classifier 2 Logistic loss
w = cp.Variable(X_train.shape[1])
b = cp.Variable()
equation = X_train @ w + b

obj_fn = cp.sum(cp.logistic(-cp.multiply( y_train, equation )))
classifier2 = cp.Problem(cp.Minimize(obj_fn))
# obj_value = classifier2.solve(verbose=True)
obj_value = classifier2.solve()

y_pred2 = np.sign(X_test @ w.value + b.value)

print("Accuracy of Classifier using logistic loss =", accuracy_score(y_test,␣
↪y_pred2))
```

```
Accuracy of Classifier using logistic loss = 0.8309859154929577
```

[31]:
```
# Classifier 3 Hinge loss
w = cp.Variable(X_train.shape[1])
b = cp.Variable()
equation = X_train @ w + b

obj_fn = cp.sum(cp.maximum(0, 1 - cp.multiply(y_train, equation)))
classifier3 = cp.Problem(cp.Minimize(obj_fn))
obj_value = classifier3.solve()

y_pred3 = np.sign(X_test @ w.value + b.value)

print("Accuracy of Classifier using hinge loss =", accuracy_score(y_test,␣
↪y_pred3))
```

```
Accuracy of Classifier using hinge loss = 0.8309859154929577
```

[32]:
```
y_test
```

[32]:
```
array([-1,  1,  1,  1, -1,  1,  1,  1, -1,  1,  1,  1, -1, -1,  1, -1, -1,
        1, -1,  1, -1,  1,  1,  1,  1, -1, -1, -1,  1, -1, -1, -1,  1, -1,
        1, -1,  1,  1, -1,  1, -1,  1,  1,  1, -1,  1, -1, -1, -1, -1, -1,
        1,  1, -1, -1,  1, -1,  1,  1, -1, -1,  1, -1,  1,  1,  1, -1, -1,
       -1, -1,  1])
```

[ ]:
```

```