Question 1.3 - 2

In [1]:
```python
import numpy as np
def f_objective(theta, X, y, l2_param):
    '''
    Args:
        theta: 1D numpy array of size num_features
        X: 2D numpy array of size (num_instances, num_features)
        y: 1D numpy array of size num_instances
        l2_param: regularization parameter

    Returns:
        objective: scalar value of objective function
    '''
    features=X.shape[1]
    instances=X.shape[0]
    loss=0
    for i in range(instances):
        loss+=np.logaddexp(0,-y[i]*np.dot(theta,X[i]))
    loss=loss/instances
    reg=l2_param*(np.dot(theta,theta))
    return(loss+reg)
```

Question 1.3 - 3

In [2]:
```python
from scipy.optimize import minimize
import pandas as pd
from functools import partial
def fit_logistic_reg(X, y, objective_function, l2_param):
    '''
    Args:
        X: 2D numpy array of size (num_instances, num_features)
        y: 1D numpy array of size num_instances
        objective_function: function returning the value of the objective
        l2_param: regularization parameter

    Returns:
        optimal_theta: 1D numpy array of size num_features
    '''
    features= X.shape[1]
    obj_func = partial(objective_function, X = X, y=y, l2_param = l2_param)
    w_0 = np.ones(features)
    w_ = minimize(obj_func, w_0).x
    return w_
```

In [3]:
```python
x_train = pd.read_csv('/Users/parthvi/Downloads/X_train.txt')
x_val = pd.read_csv('/Users/parthvi/Downloads/X_val.txt')
y_train = pd.read_csv('/Users/parthvi/Downloads/y_train.txt')
y_val = pd.read_csv('/Users/parthvi/Downloads/y_val.txt')
```

In [4]:
```python
from sklearn.preprocessing import StandardScaler
X_train = StandardScaler().fit_transform(x_train)
X_val = StandardScaler().fit_transform(x_val)
X_train = np.hstack((X_train, np.ones((X_train.shape[0],1))))
X_val = np.hstack((X_val, np.ones((X_val.shape[0],1))))
y_train[y_train == 0] = -1
y_val[y_val == 0] = -1

#y_train = y_train.astype(int)
#y_train = np.array(y_train)
```

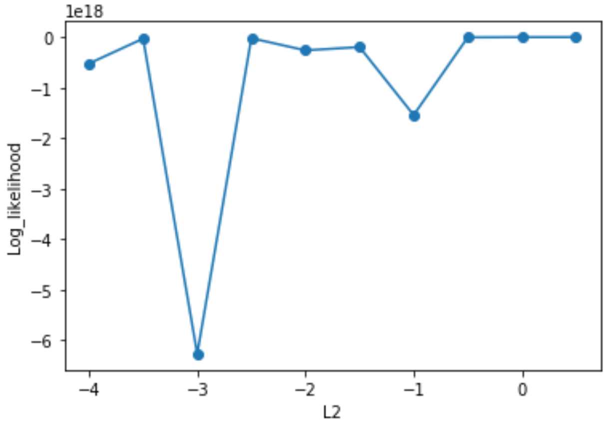
In [5]:
```python
y_trainn  = y_train.to_numpy()
y_train = y_trainn.reshape(y_train.shape[0])
y_val   = y_val.to_numpy()
y_val = y_val.reshape(y_val.shape[0])
```

In [6]:
```
theta = fit_logistic_reg(X_train, y_train, f_objective, l2_param=1)
```

Question 1.3 - 4

In [32]:
```
L2=list([i for i in np.arange(-4,1,0.5)])
log_likelihood_list=[]
for l2reg in L2:
    theta = fit_logistic_reg(X_train, y_train, f_objective, l2_param=l2reg)
    result = f_objective(theta,X_val,y_val, l2reg)
    log_likelihood_list.append(result)
```

In [33]:
```
import matplotlib.pyplot as plt
plt.plot(L2,log_likelihood_list, 'o-')
plt.xlabel('L2')
plt.ylabel('Log_likelihood')
plt.show()
```



The lowest log_likelihood is achieved at L2 = -3

In [ ]:

DSGA 1003 - HW 5 -

Q. 2 Bayesian Logistic Regression with Gaussian priors.

1) posterior density $p(w|D)$

$\rightarrow$ $p(w|D) = p(w|x, y)$

using Bayes rule,

$$p(y|x, w) \cdot p(w) / p(y)$$

$$\propto p(y|x, w)\, p(w)$$

$$\Rightarrow \prod_{i=1}^{N} \sigma(w^T x_i)^{y_i} (1 - \sigma(w^T x_i))^{1-y_i}$$

Negative log likelihood:

$$N\log L(w) = -\sum_{i=1}^{N} y^{(i)} \log \sigma(w^T x_i) + (1-y^{i}) \log (1 - \sigma(w^T x_i))$$

$$\left\{ \left[\text{where } \sigma = \frac{1}{1+e^{-z}}\right] \boxed{P(Y = 1 | X = x) = \sigma(z)} \right\}$$

$$p(w|D) = \frac{P(D|w) \cdot P(w)}{P(D)} = \frac{\exp(-NLL_D(w))\, P(w)}{P(D)}$$

$\rightarrow$ prior density $p(w)$,

Q. 1. 2

1. Decision boundary of logistic regression is a set of all points x that satisfy:

$$P(y=1|x) = P(y=0|x) = \frac{1}{2}$$

$$\therefore P(y=1|x) = \frac{1}{1+e^{-\theta^T x_+}}$$

$$\therefore \frac{1}{1+e^{-\theta^T x_+}} = \frac{1}{2}$$

$$\therefore \theta^T x_+ = 0$$

$$\therefore \theta_1 x_1 + \dots \theta_d x_d = 0$$

for x_1, x_2

$$\theta_1 x_1 + \theta_2 x_2 = 0$$

$$\therefore x_2 = \frac{-\theta_1 x_1}{\theta_2}$$

$$\therefore \cancel{\theta} = \boxed{x_2 = x^T \theta} \quad \text{where } \theta = \frac{-\theta_1}{\theta_2}$$

In general: $x = x^T \theta$

In our case $\boxed{\theta = w}$

$$\therefore x = x^T w$$

1.2

(a) $L(c\bar{w}) = \sum_{i=1}^{n} y_i \log p(x_i) + (1-y_i)\log 1-p(x_i)$

$= \sum_{i=1}^{n} \log 1-p(x_i) + \sum_{i=1}^{n} y_i (x_i \cdot cw)$

$= \sum_{i=1}^{n} -\log 1 + e^{cwx_i} + \sum_{i=1}^{n} y_i (x_i \cdot cw)$

$\therefore \frac{\partial L}{\partial c} \Rightarrow - \sum_{i=1}^{n} \frac{1}{1+e^{cwx_i}} \cdot e^{cwx_i} wx_{ij} + \sum_{i=1}^{n} y_i x_{ij} w$

$= \sum_{i=1}^{n} (y_i - p(wx_i; cw)) x_{ij} w$

Now, we can see that this is not a closed form solution. We can only approximate it numerically. As $c \uparrow$ & $p(x_i w; c \cdot w) \uparrow$

Q. 1

1.1.

ERM with logistic loss:

$F_{score} = \{x \rightarrow x^T w \mid w \in \mathbb{R}^d\}$

$l_{logistic} = \log(1+\exp(-y w^T x))$

$\rightarrow \hat{R}_n(w) = \frac{1}{n} \sum_{i=1}^{n} l_{logistic}(y_i w^T x_i)$

$= \frac{1}{n} \sum_{i=1}^{n} \log(1+\exp(\underbrace{-y_i w^T x_i}_{①}))$

MLE with bernoulli

$\rightarrow p(y=1 \mid x; w) = \frac{1}{1+\exp(-x^T w)}$

$NLL(w) = \sum_{i=1}^{n} [-y_i \log \sigma(w^T x_i)] + (y_i - 1)\log(1-\sigma(w^T x_i))$

st $\hat{w}_*$ minimize $NLL(w)$; $x \rightarrow \sigma(x^T \hat{w}_*)$
is Max. likelihood prediction f^n.

$$NLL(w) = \sum_{i=1}^{n} [-\log \sigma(w^T x_i)]$$

$$= \sum_{i=1}^{n} \log(1 + \exp(-y_i w^T x_i))$$

$$= n \hat{R}_n(w) = ERM$$

$\therefore$ The approaches are equivalent.

Q. 1.3

1. $J_{logistic}(w) = \hat{R}_n(w) + \lambda \|w\|^2$

→ We know, $\lambda \|w\|^2$ is a convex function.
We also know exponential f^ns are convex.
& log of a convex function is convex.

$\therefore$ The objective f^n is a convex f^n.
$\because \nabla^2 J(w)$ is ≥ 0

Q. 2

2 No, this is not a conjugate prior to the likelihood given by logistic regression.

$\because$ Product of gaussian prior on w & likelihood $\neq$ posterior

3 $-\log(P(w|D)) = NLL - \log P(w) + \log P(D)$

We know,

$$NLL = n\hat{R}_n$$

$$\therefore \quad -\log P(w) = -\frac{1}{2}\log|(2\pi\Sigma)| + \underbrace{\frac{1}{2} w^T \Sigma^{-1} w}_{= n\lambda w^T w}$$

$$\therefore \quad \boxed{\Sigma = \frac{1}{2n\lambda} I}$$

Q. 4 $\dfrac{1}{2n\lambda} = 1 \qquad (\text{if } \Sigma = I)$

$$\therefore \quad \boxed{\lambda = \frac{1}{2n}}$$

Q.3

1. $p(Z=H|\theta_1) = \theta_1$

$p(X=H|Z=H, \theta_2) = \theta_2$

$P(X=T|Z=T) = 1$.

show that: $p(X=H|\theta_1, \theta_2) = \theta_1\theta_2$

$\rightarrow P(X=H) = P(X=H|Z=H)\,P(Z=H) + P(X=H|Z=T)\cdot P(Z=T)$

$= \underbrace{P(X=H|Z=H)}_{\theta_2}\,\underbrace{P(Z=H)}_{\theta_1} + \underbrace{P(X=H|Z=T)}_{0}\cdot\underbrace{P(Z=T)}_{1-\theta_1}$

$\Rightarrow \theta_2 \cdot \theta_1$

2. $N_r = n_h + n_t$

① $P(D|\theta_1, \theta_2)$ using binomial pmf

$\rightarrow P(D|\theta_1, \theta_2) = P(D|\theta_1) \cdot P(D|\theta_2)$

$= \left({}^{n_r}C_{n_h}\, \theta_1^{n_h} (1-\theta_1)^{n_t}\right)\left({}^{n_r}C_{n_h}\, \theta_2^{n_h} (1-\theta_2)^{n_t}\right)$

$$\log\left(P(D \mid \theta_1, \theta_2)\right)$$

$$= \log\left[\left({}^{n_r}C_{n_h}\, \theta_1^{n_h} (1-\theta_1)^{n_t}\right)\left({}^{n_r}C_{n_h}\, \theta_2^{n_h} (1-\theta_2)^{n_h}\right)\right]$$

$$= \log\left({}^{n_r}C_{n_h}\, \theta_1^{n_h} (1-\theta_1)^{n_t}\right) + \log\left({}^{n_r}C_{n_h}\, \theta_2^{n_h} (1-\theta_2)^{n_t}\right)$$

$$= \log {}^{n_r}C_{n_h} + n_h \log \theta_1 + n_t \log(1-\theta_1)$$

$$+ \log\left({}^{n_r}C_{n_h}\right) + n_h \log \theta_2 + n_t \log(1-\theta_2)$$

Taking ∇ w.r.t θ_1:

$$\frac{n_h}{\theta_1} + \frac{n_t}{1-\theta_1}(-1) = 0$$

$$\therefore \frac{n_h}{\theta_1} = \frac{n_t}{1-\theta_1}$$

$$\therefore n_h - \theta_1 n_h = \theta_1 n_t$$

$$\therefore n_h = \theta_1 (n_h + n_t)$$

$$\therefore \theta_1 = \frac{n_h}{n_h + n_t}$$

Similarly w.r.t. θ_2

$$\theta_2 = \frac{n_h}{n_h + n_t}$$

Yes

3. $L(\theta_1, \theta_2) = p(D_r, D_c \mid \theta_1, \theta_2)$

$= P(D_r \mid \theta_1, \theta_2) * P(D_c \mid \theta_1, \theta_2)$

$P(D_c \mid \theta_1, \theta_2) = P(D_c \mid \theta_1) \sim$ Binomial

$= {}^{n}C_{\theta_1} \; \theta_1^{c_h} (1-\theta_1)^{c_t}$