In [1]:

```
import matplotlib.pyplot as plt
import numpy as np
import numpy.random
import numpy.linalg
import scipy.io
import scipy.stats
import sklearn.metrics

# setup plotting
from IPython import get_ipython
import psutil
inTerminal = not "IPKernelApp" in get_ipython().config
inJupyterNb = any(filter(lambda x: x.endswith("jupyter-notebook"), psutil.Process().parent().cmdline()))
get_ipython().run_line_magic("matplotlib", "" if inTerminal else "notebook" if inJupyterNb else "widget")
def nextplot():
    if inTerminal:
        plt.clf()     # this clears the current plot
    else:
        plt.figure()  # this creates a new plot
```

Load the data

In [2]:

```
data = scipy.io.loadmat("data/spamData.mat")
X = data["Xtrain"]
N = X.shape[0]
D = X.shape[1]
Xtest = data["Xtest"]
Ntest = Xtest.shape[0]
y = data["ytrain"].squeeze().astype(int)
ytest = data["ytest"].squeeze().astype(int)

features = np.array(
    [
        "word_freq_make",
        "word_freq_address",
        "word_freq_all",
        "word_freq_3d",
        "word_freq_our",
        "word_freq_over",
        "word_freq_remove",
        "word_freq_internet",
        "word_freq_order",
        "word_freq_mail",
        "word_freq_receive",
        "word_freq_will",
        "word_freq_people",
        "word_freq_report",
        "word_freq_addresses",
        "word_freq_free",
        "word_freq_business",
        "word_freq_email",
        "word_freq_you",
        "word_freq_credit",
        "word_freq_your",
        "word_freq_font",
        "word_freq_000",
        "word_freq_money",
        "word_freq_hp",
        "word_freq_hpl",
        "word_freq_george",
        "word_freq_650",
        "word_freq_lab",
        "word_freq_labs",
        "word_freq_telnet",
        "word_freq_857",
        "word_freq_data",
        "word_freq_415",
        "word_freq_85",
        "word_freq_technology",
        "word_freq_1999",
        "word_freq_parts",
        "word_freq_pm",
        "word_freq_direct",
        "word_freq_cs",
        "word_freq_meeting",
        "word_freq_original",
        "word_freq_project",
        "word_freq_re",
        "word_freq_edu",
        "word_freq_table",
        "word_freq_conference",
        "char_freq_;",
        "char_freq_(",
        "char_freq_[",
        "char_freq_!",
        "char_freq_$",
        "char_freq_#",
```

```
        "capital_run_length_average",
        "capital_run_length_longest",
        "capital_run_length_total",
    ]
)
```

1. Dataset Statistics

In [3]:
```
# look some dataset statistics
scipy.stats.describe(X)
```

Out[3]:
```
DescribeResult(nobs=3065, minmax=(array([0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
       0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
       0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
       0., 0., 0., 1., 1., 1.]), array([4.5400e+00, 1.4280e+01, 5.1000e+00, 4.2810e+01, 9.0900e+00,
       3.5700e+00, 7.2700e+00, 1.1110e+01, 3.3300e+00, 1.8180e+01,
       2.0000e+00, 9.6700e+00, 5.5500e+00, 5.5500e+00, 2.8600e+00,
       1.0160e+01, 7.1400e+00, 9.0900e+00, 1.8750e+01, 6.3200e+00,
       1.1110e+01, 1.7100e+01, 5.4500e+00, 9.0900e+00, 2.0000e+01,
       1.4280e+01, 3.3330e+01, 4.7600e+00, 1.4280e+01, 4.7600e+00,
       4.7600e+00, 4.7600e+00, 1.8180e+01, 4.7600e+00, 2.0000e+01,
       7.6900e+00, 6.8900e+00, 7.4000e+00, 9.7500e+00, 4.7600e+00,
       7.1400e+00, 1.4280e+01, 3.5700e+00, 2.0000e+01, 2.1420e+01,
       1.6700e+01, 2.1200e+00, 1.0000e+01, 4.3850e+00, 9.7520e+00,
       4.0810e+00, 3.2478e+01, 6.0030e+00, 1.9829e+01, 1.1025e+03,
       9.9890e+03, 1.5841e+04])), mean=array([1.10818923e-01, 2.28486134e-01, 2.74153344e-01, 6.29690049e-02,
       3.17787928e-01, 9.57553018e-02, 1.13546493e-01, 1.07216966e-01,
       8.89233279e-02, 2.41719413e-01, 5.81305057e-02, 5.37432300e-01,
       9.26231648e-02, 4.96639478e-02, 5.07210440e-02, 2.35334421e-01,
       1.47197390e-01, 1.86600326e-01, 1.66121044e+00, 7.63066884e-02,
       8.19592170e-01, 1.22727569e-01, 1.02006525e-01, 8.90799347e-02,
       5.29800979e-01, 2.62071778e-01, 7.71507341e-01, 1.14323002e-01,
       1.09487765e-01, 9.92952692e-02, 6.28156607e-02, 4.90342577e-02,
       9.27471452e-02, 4.96019576e-02, 1.02156607e-01, 9.93050571e-02,
       1.43285481e-01, 1.24274062e-02, 7.55921697e-02, 6.60456770e-02,
       4.63360522e-02, 1.32176183e-01, 4.88580750e-02, 7.11876020e-02,
       3.06590538e-01, 1.79794454e-01, 5.28874388e-03, 3.13768352e-02,
       3.79543230e-02, 1.38396411e-01, 1.81830343e-02, 2.65470799e-01,
       7.91275693e-02, 5.34218597e-02, 4.90062936e+00, 5.26750408e+01,
       2.82203915e+02]), variance=array([1.07094140e-01, 1.88742036e+00, 2.34317437e-01, 1.78161723e+00,
       4.40325719e-01, 6.79193461e-02, 1.39844435e-01, 1.72001423e-01,
       6.97247542e-02, 4.69800274e-01, 3.58302179e-02, 7.59167719e-01,
       9.28365241e-02, 8.26118648e-02, 7.00470321e-02, 4.29393369e-01,
       2.00636301e-01, 2.92991898e-01, 3.18992370e+00, 1.65626303e-01,
       1.44315254e+00, 1.01505046e+00, 1.19749530e-01, 1.43862796e-01,
       2.45800502e+00, 7.38036013e-01, 1.13920029e+01, 2.31010973e-01,
       4.31507668e-01, 1.90528093e-01, 1.24671084e-01, 1.07425177e-01,
       2.95159161e-01, 1.07745599e-01, 3.08154062e-01, 1.67896547e-01,
       1.85791650e-01, 4.34829439e-02, 1.42525114e-01, 1.16865102e-01,
       1.50361473e-01, 6.09903912e-01, 5.73945833e-02, 3.19259425e-01,
       1.01935877e+00, 8.17471270e-01, 4.63438951e-03, 7.50333517e-02,
       5.54612799e-02, 7.77968333e-02, 1.48045497e-02, 7.59181612e-01,
       6.74541224e-02, 2.69600271e-01, 7.42311765e+02, 4.86573219e+04,
       3.68952901e+05]), skewness=array([ 5.92257918,  9.5555492 ,  2.94110789, 27.15035267,  4.22000271,
        4.55490419,  6.21454549, 10.63604439,  4.44795353,  9.63368819,
        5.1601559 ,  3.12797362,  7.99555783, 10.07103212,  6.44051978,
        5.9017492 ,  5.71193665,  5.63845456,  1.6918398 ,  8.05102821,
        2.36131511,  9.70708774,  5.74851972, 13.62929854,  5.51200726,
        5.77490458,  5.72163481,  5.84582426, 11.30526457,  6.67894971,
        8.78006633, 10.35563132, 16.1291286 , 10.31146394, 17.98980105,
        7.86085564,  5.29526945, 27.69555992, 10.51869112,  9.12514394,
       12.60532735,  9.42688905,  7.88762618, 19.69945392,  9.63372543,
        8.97501221, 18.94255005, 20.98217881, 14.12336521, 16.36382061,
       21.32440567, 21.32959254, 10.88427173, 26.25786993, 27.34951229,
       31.14016596,  9.80477376]), kurtosis=array([  51.71558405,   93.89016173,   13.18839908,  785.40163828,
         28.69487647,   31.20576951,   66.53150801,  198.68010939,
         28.29530115,  185.40607771,   34.48800593,   15.18712484,
        109.66544541,  138.05561341,   44.19188958,   55.62892   ,
         47.49151277,   52.75647121,    6.32523058,   77.87379384,
          8.48736408,  103.7022867 ,   49.37553046,  272.09125904,
         42.43992409,   49.41302953,   33.63974328,   39.86629858,
        166.19735746,   53.12216402,   91.72439904,  124.79234055,
        433.42661801,  123.97955409,  555.16708959,   86.72460731,
         43.92486688,  865.39968623,  181.33012173,  100.87592785,
        189.11563172,  111.21705016,   81.96093958,  567.75150773,
        147.5283386 ,  107.79164424,  445.8361165 ,  634.57001982,
        228.75884956,  499.07842266,  588.19774644,  688.05527222,
        184.31757803,  851.48819158,  954.59095344, 1348.49464105,
        183.78053905]))
```

In [4]:
```
scipy.stats.describe(y)
```

Out[4]:
```
DescribeResult(nobs=3065, minmax=(0, 1), mean=0.39738988580750406, variance=0.23954932085067238, skewness=0.419366324
78193103, kurtosis=-1.824131885638896)
```

In [5]:
```
# plot the distribution of all features
nextplot()
```

```
densities = [scipy.stats.gaussian_kde(X[:, j]) for j in range(D)]
xs = np.linspace(0, np.max(X), 200)
for j in range(D):
    plt.plot(xs, densities[j](xs), label=j)
plt.legend(ncol=5)
```

Out[5]:
```
<matplotlib.legend.Legend at 0x7fc0ae37f400>
```

In [6]:
```
# this plots is not really helpful; go now explore further
# YOUR CODE HERE
nextplot()
densities = [scipy.stats.gaussian_kde(X[:, j]) for j in range(D)]
xs = np.linspace(-.3,.3)
for j in range(D):
    plt.plot(xs, densities[j](xs), label=j)
plt.legend(ncol=5)
```

Out[6]:
```
<matplotlib.legend.Legend at 0x7fc0b07db970>
```

In [7]:
```
import pandas as pd
df = pd.DataFrame(X)
print(df.describe())
```

```
                0            1            2            3            4   \
count  3065.000000  3065.000000  3065.000000  3065.000000  3065.000000
mean      0.110819     0.228486     0.274153     0.062969     0.317788
std       0.327252     1.373834     0.484063     1.334772     0.663570
min       0.000000     0.000000     0.000000     0.000000     0.000000
25%       0.000000     0.000000     0.000000     0.000000     0.000000
50%       0.000000     0.000000     0.000000     0.000000     0.000000
75%       0.000000     0.000000     0.410000     0.000000     0.390000
max       4.540000    14.280000     5.100000    42.810000     9.090000

                5            6            7            8            9   ...  \
count  3065.000000  3065.000000  3065.000000  3065.000000  3065.000000  ...
mean      0.095755     0.113546     0.107217     0.088923     0.241719  ...
std       0.260613     0.373958     0.414731     0.264054     0.685420  ...
min       0.000000     0.000000     0.000000     0.000000     0.000000  ...
25%       0.000000     0.000000     0.000000     0.000000     0.000000  ...
50%       0.000000     0.000000     0.000000     0.000000     0.000000  ...
75%       0.000000     0.000000     0.000000     0.000000     0.160000  ...
max       3.570000     7.270000    11.110000     3.330000    18.180000  ...

               47           48           49           50           51  \
count  3065.000000  3065.000000  3065.000000  3065.000000  3065.000000
mean      0.031377     0.037954     0.138396     0.018183     0.265471
std       0.273922     0.235502     0.278921     0.121674     0.871310
min       0.000000     0.000000     0.000000     0.000000     0.000000
25%       0.000000     0.000000     0.000000     0.000000     0.000000
50%       0.000000     0.000000     0.066000     0.000000     0.000000
75%       0.000000     0.000000     0.191000     0.000000     0.315000
max      10.000000     4.385000     9.752000     4.081000    32.478000

               52           53           54           55           56
count  3065.000000  3065.000000  3065.000000  3065.000000  3065.000000
mean      0.079128     0.053422     4.900629    52.675041   282.203915
std       0.259719     0.519230    27.245399   220.584047   607.414933
min       0.000000     0.000000     1.000000     1.000000     1.000000
25%       0.000000     0.000000     1.600000     6.000000    35.000000
50%       0.000000     0.000000     2.280000    15.000000    97.000000
75%       0.056000     0.000000     3.737000    43.000000   272.000000
max       6.003000    19.829000  1102.500000  9989.000000  15841.000000

[8 rows x 57 columns]
```

In [8]:
```
# Let's compute z-scores; create two new variables Xz and Xtestz.
Xz = (X - np.mean(X,axis=0))/np.std(X,axis=0)
Xtestz = (Xtest - np.mean(X,axis=0))/np.std(X,axis=0)
```

In [9]:
```
# Let's check. Xz and Xtestz refer to the normalized datasets just created. We
# will use them throughout.
np.mean(Xz, axis=0)  # should be all 0
np.var(Xz, axis=0)  # should be all 1
np.mean(Xtestz, axis=0)  # what do you get here?
np.var(Xtestz, axis=0)

np.sum(Xz ** 3)  # should be: 1925261.15
```

Out[9]:
```
1925261.1560010156
```

In [10]:
```
# Explore the normalized data
# YOUR CODE HERE
```

```
nextplot()
densities = [scipy.stats.gaussian_kde(Xz[:, j]) for j in range(D)]
xsz = np.linspace(-5,5)
for j in range(D):
    plt.plot(xsz, densities[j](xsz), label=j)
```

2. Maximum Likelihood Estimation

In [11]:

```
import numpy as np
import matplotlib.pyplot as plt
def sigmoid(x) :
    return 1/(1+np.exp(-x))

x = np.arange(-5.0,5.0,0.1)
y1 = sigmoid(x+0.5)
y2 = sigmoid(x+1)
y3 = sigmoid(x+1.5)

nextplot()
plt.plot(x,y1,'r', linestyle = '--') #x+0.5
plt.plot(x,y2,'g') #x+1
plt.plot(x,y3,'b', linestyle = '--') #x+1.5
plt.plot([0,0],[1.0,0.0],':') #add dotted line in the middle
plt.title('Sigmoid Function')
plt.show()
plt.savefig('Figure2.1.png')
```

Helper functions

In [12]:

```
def logsumexp(x):
    """Computes log(sum(exp(x)).

    Uses offset trick to reduce risk of numeric over- or underflow. When x is a
    1D ndarray, computes logsumexp of its entries. When x is a 2D ndarray,
    computes logsumexp of each column.

    Keyword arguments:
    x : a 1D or 2D ndarray
    """
    offset = np.max(x, axis=0)
    return offset + np.log(np.sum(np.exp(x - offset), axis=0))
```

In [13]:

```
# Define the logistic function. Make sure it operates on both scalars
# and vectors.
def sigma(x):
    # YOUR CODE HERE
    logistic_function = 1/(1+np.exp(-x))
    return logistic_function
```

In [14]:

```
# this should give:
# [0.5, array([0.26894142, 0.5, 0.73105858])]
[sigma(0), sigma(np.array([-1, 0, 1]))]
```

Out[14]:

```
[0.5, array([0.26894142, 0.5       , 0.73105858])]
```

In [15]:

```
# Define the logarithm of the logistic function. Make sure it operates on both
# scalars and vectors. Perhaps helpful: isinstance(x, np.ndarray).
def logsigma(x):
    # YOUR CODE HERE
    log_logistic_function = np.log(sigma(x))
    return log_logistic_function
```

In [16]:

```
# this should give:
# [-0.69314718055994529, array([-1.31326169, -0.69314718, -0.31326169])]
[logsigma(0), logsigma(np.array([-1, 0, 1]))]
```

Out[16]:

```
[-0.6931471805599453, array([-1.31326169, -0.69314718, -0.31326169])]
```

2b Log-likelihood and gradient

In [17]:

```
def l(y, X, w):
    """Log-likelihood of the logistic regression model.

    Parameters
    ----------
```

```
    y : ndarray of shape (N,)
        Binary labels (either 0 or 1).
    X : ndarray of shape (N,D)
        Design matrix.
    w : ndarray of shape (D,)
        Weight vector.
    """
    # YOUR CODE HERE
    z = np.dot(X,w)
    return np.sum(y*z - np.log(1+np.exp(z)))
```

In [18]:
```
# this should give:
# -47066.641667825766
l(y, Xz, np.linspace(-5, 5, D))
```

Out[18]:
```
-47066.64166782577
```

In [19]:
```
def dl(y, X, w):
    """Gradient of the log-likelihood of the logistic regression model.

    Parameters
    ----------
    y : ndarray of shape (N,)
        Binary labels (either 0 or 1).
    X : ndarray of shape (N,D)
        Design matrix.
    w : ndarray of shape (D,)
        Weight vector.

    Returns
    -------
    ndarray of shape (D,)
    """
    # YOUR CODE HERE
    z = np.dot(X,w)
    prediction = sigma(z)
    errors = y - prediction
    gradients = np.dot(X.T,errors)
    return gradients
```

In [20]:
```
# this should give:
# array([  551.33985842,   143.84116318,   841.83373606,   156.87237578,
#          802.61217579,   795.96202907,   920.69045803,   621.96516752,
#          659.18724769,   470.81259805,   771.32406968,   352.40325626,
#          455.66972482,   234.36600888,   562.45454038,   864.83981264,
#          787.19723703,   649.48042176,   902.6478154 ,   544.00539886,
#         1174.78638035,   120.3598967 ,   839.61141672,   633.30453444,
#         -706.66815087,  -630.2039816 ,  -569.3451386 ,  -527.50996698,
#         -359.53701083,  -476.64334832,  -411.60620464,  -375.11950586,
#         -345.37195689,  -376.22044258,  -407.31761977,  -456.23251936,
#         -596.86960184,  -107.97072355,  -394.82170044,  -229.18125598,
#         -288.46356547,  -362.13402385,  -450.87896465,  -277.03932676,
#         -414.99293368,  -452.28771693,  -167.54649092,  -270.9043748 ,
#         -252.20140951,  -357.72497343,  -259.12468742,   418.35938483,
#          604.54173228,    43.10390907,   152.24258478,   378.16731033,
#          416.12032881])
dl(y, Xz, np.linspace(-5, 5, D))
```

Out[20]:
```
array([ 551.33985842,  143.84116318,  841.83373606,  156.87237578,
        802.61217579,  795.96202907,  920.69045803,  621.96516752,
        659.18724769,  470.81259805,  771.32406968,  352.40325626,
        455.66972482,  234.36600888,  562.45454038,  864.83981264,
        787.19723703,  649.48042176,  902.6478154 ,  544.00539886,
       1174.78638035,  120.3598967 ,  839.61141672,  633.30453444,
       -706.66815087, -630.2039816 , -569.3451386 , -527.50996698,
       -359.53701083, -476.64334832, -411.60620464, -375.11950586,
       -345.37195689, -376.22044258, -407.31761977, -456.23251936,
       -596.86960184, -107.97072355, -394.82170044, -229.18125598,
       -288.46356547, -362.13402385, -450.87896465, -277.03932676,
       -414.99293368, -452.28771693, -167.54649092, -270.9043748 ,
       -252.20140951, -357.72497343, -259.12468742,  418.35938483,
        604.54173228,   43.10390907,  152.24258478,  378.16731033,
        416.12032881])
```

2c Gradient descent

In [21]:
```
# you don't need to modify this function
def optimize(obj_up, theta0, nepochs=50, eps0=0.01, verbose=True):
    """Iteratively minimize a function.

    We use it here to run either gradient descent or stochastic gradient
    descent, using arbitrarly optimization criteria.

    Parameters
    ----------
    obj_up  : a tuple of form (f, update) containing two functions f and update.
```

```
            f(theta) computes the value of the objective function.
            update(theta,eps) performs an epoch of parameter update with step size
            eps and returns the result.
    theta0  : ndarray of shape (D,)
            Initial parameter vector.
    nepochs : int
            How many epochs (calls to update) to run.
    eps0    : float
            Initial step size.
    verbose : boolean
            Whether to print progress information.

    Returns
    -------
    A triple consisting of the fitted parameter vector, the values of the
    objective function after every epoch, and the step sizes that were used.
    """

    f, update = obj_up

    # initialize results
    theta = theta0
    values = np.zeros(nepochs + 1)
    eps = np.zeros(nepochs + 1)
    values[0] = f(theta0)
    eps[0] = eps0

    # now run the update function nepochs times
    for epoch in range(nepochs):
        if verbose:
            print(
                "Epoch {:3d}: f={:10.3f}, eps={:10.9f}".format(
                    epoch, values[epoch], eps[epoch]
                )
            )
        theta = update(theta, eps[epoch])

        # we use the bold driver heuristic
        values[epoch + 1] = f(theta)
        if values[epoch] < values[epoch + 1]:
            eps[epoch + 1] = eps[epoch] / 2.0
        else:
            eps[epoch + 1] = eps[epoch] * 1.05

    # all done
    if verbose:
        print("Result after {} epochs: f={}".format(nepochs, values[-1]))
    return theta, values, eps
```

In [22]:

```
# define the objective and update function for one gradient-descent epoch for
# fitting an MLE estimate of logistic regression with gradient descent (should
# return a tuple of two functions; see optimize)
def gd(y, X):
    def objective(w):
        # YOUR CODE HERE
        p_i1 = logsigma(np.dot(X,w))
        p_i0 = logsigma(-np.dot(X,w))
        objective = -np.sum(y*p_i1 + (1-y)*p_i0)
        return objective

    def update(w, eps):
        # YOUR CODE HERE
        gradients = dl(y,X,w)
        w += eps * gradients
        return w

    return (objective, update)
```

In [23]:

```
# this should give
# [47066.641667825766,
#  array([  4.13777838e+01,  -1.56745627e+01,   5.75882538e+01,
#           1.14225143e+01,   5.54249703e+01,   5.99229049e+01,
#           7.11220141e+01,   4.84761728e+01,   5.78067289e+01,
#           4.54794720e+01,   7.14638492e+01,   1.51369386e+01,
#           3.36375739e+01,   2.15061217e+01,   5.78014255e+01,
#           6.72743066e+01,   7.00829312e+01,   5.29328088e+01,
#           6.16042473e+01,   5.50018510e+01,   8.94624817e+01,
#           2.74784480e+01,   8.51763599e+01,   5.60363965e+01,
#          -2.55865589e+01,  -1.53788213e+01,  -4.67015412e+01,
#          -2.50356570e+00,  -3.85357592e+00,  -2.21819155e+00,
#           3.32098671e+00,   3.86933390e+00,  -2.00309898e+01,
#           3.84684492e+00,  -2.19847927e-01,  -1.29775457e+00,
#          -1.28374302e+01,  -2.78303173e+00,  -5.61671182e+00,
#           1.73657121e+01,  -6.81197570e+00,  -1.20249002e+01,
#           2.65789491e+00,  -1.39557852e+01,  -2.01135653e+01,
#          -2.72134051e+01,  -9.45952961e-01,  -1.02239111e+01,
#           1.52794293e-04,  -5.18938123e-01,  -3.19717561e+00,
#           4.62953437e+01,   7.87893022e+01,   1.88618651e+01,
#           2.85195027e+01,   5.04698358e+01,   6.41240689e+01])
```

```
f, update = gd(y, Xz)
[f(np.linspace(-5, 5, D)), update(np.linspace(-5, -5, D), 0.1)]
```

Out[23]:

```
[47066.64166782577,
 array([ 4.13777838e+01, -1.56745627e+01,  5.75882538e+01,  1.14225143e+01,
         5.54249703e+01,  5.99229049e+01,  7.11220141e+01,  4.84761728e+01,
         5.78067289e+01,  4.54794720e+01,  7.14638492e+01,  1.51369386e+01,
         3.36375739e+01,  2.15061217e+01,  5.78014255e+01,  6.72743066e+01,
         7.00829312e+01,  5.29328088e+01,  6.16042473e+01,  5.50018510e+01,
         8.94624817e+01,  2.74784480e+01,  8.51763599e+01,  5.60363965e+01,
        -2.55865589e+01, -1.53788213e+01, -4.67015412e+01, -2.50356570e+00,
        -3.85357592e+00, -2.21819155e+00,  3.32098671e+00,  3.86933390e+00,
        -2.00309898e+01,  3.84684492e+00, -2.19847927e-01, -1.29775457e+00,
        -1.28374302e+01, -2.78303173e+00, -5.61671182e+00,  1.73657121e+01,
        -6.81197570e+00, -1.20249002e+01,  2.65789491e+00, -1.39557852e+01,
        -2.01135653e+01, -2.72134051e+01, -9.45952961e-01, -1.02239111e+01,
         1.52794293e-04, -5.18938123e-01, -3.19717561e+00,  4.62953437e+01,
         7.87893022e+01,  1.88618651e+01,  2.85195027e+01,  5.04698358e+01,
         6.41240689e+01])]
```

In [24]:

```
# you can run gradient descent!
numpy.random.seed(0)
w0 = np.random.normal(size=D)
wz_gd, vz_gd, ez_gd = optimize(gd(y, Xz), w0, nepochs=500)
```

```
Epoch   0: f=  6636.208, eps=0.010000000
Epoch   1: f=  4216.957, eps=0.010500000
Epoch   2: f=  2657.519, eps=0.011025000
Epoch   3: f=  1926.135, eps=0.011576250
Epoch   4: f=  1449.495, eps=0.012155063
Epoch   5: f=  1207.529, eps=0.012762816
Epoch   6: f=  1052.489, eps=0.013400956
Epoch   7: f=   957.275, eps=0.014071004
Epoch   8: f=   899.610, eps=0.014774554
Epoch   9: f=   882.904, eps=0.015513282
Epoch  10: f=  1017.083, eps=0.007756641
Epoch  11: f=   840.760, eps=0.008144473
Epoch  12: f=   805.649, eps=0.008551697
Epoch  13: f=   822.108, eps=0.004275848
Epoch  14: f=   746.377, eps=0.004489641
Epoch  15: f=   735.803, eps=0.004714123
Epoch  16: f=   729.780, eps=0.004949829
Epoch  17: f=   724.467, eps=0.005197320
Epoch  18: f=   719.408, eps=0.005457186
Epoch  19: f=   714.564, eps=0.005730046
Epoch  20: f=   709.932, eps=0.006016548
Epoch  21: f=   705.514, eps=0.006317375
Epoch  22: f=   701.321, eps=0.006633244
Epoch  23: f=   697.373, eps=0.006964906
Epoch  24: f=   693.728, eps=0.007313152
Epoch  25: f=   690.591, eps=0.007678809
Epoch  26: f=   688.614, eps=0.008062750
Epoch  27: f=   688.607, eps=0.008465887
Epoch  28: f=   690.854, eps=0.004232944
Epoch  29: f=   679.967, eps=0.004444591
Epoch  30: f=   678.649, eps=0.004666820
Epoch  31: f=   677.447, eps=0.004900161
Epoch  32: f=   676.292, eps=0.005145169
Epoch  33: f=   675.182, eps=0.005402428
Epoch  34: f=   674.120, eps=0.005672549
Epoch  35: f=   673.114, eps=0.005956177
Epoch  36: f=   672.177, eps=0.006253986
Epoch  37: f=   671.334, eps=0.006566685
Epoch  38: f=   670.656, eps=0.006895019
Epoch  39: f=   670.397, eps=0.007239770
Epoch  40: f=   671.342, eps=0.003619885
Epoch  41: f=   668.932, eps=0.003800879
Epoch  42: f=   668.378, eps=0.003990923
Epoch  43: f=   668.027, eps=0.004190469
Epoch  44: f=   667.720, eps=0.004399993
Epoch  45: f=   667.433, eps=0.004619993
Epoch  46: f=   667.159, eps=0.004850992
Epoch  47: f=   666.897, eps=0.005093542
Epoch  48: f=   666.650, eps=0.005348219
Epoch  49: f=   666.417, eps=0.005615630
Epoch  50: f=   666.201, eps=0.005896411
Epoch  51: f=   666.008, eps=0.006191232
Epoch  52: f=   665.858, eps=0.006500794
Epoch  53: f=   665.812, eps=0.006825833
Epoch  54: f=   666.068, eps=0.003412917
Epoch  55: f=   665.424, eps=0.003583562
Epoch  56: f=   665.290, eps=0.003762741
Epoch  57: f=   665.204, eps=0.003950878
Epoch  58: f=   665.128, eps=0.004148421
Epoch  59: f=   665.054, eps=0.004355843
Epoch  60: f=   664.982, eps=0.004573635
Epoch  61: f=   664.911, eps=0.004802316
Epoch  62: f=   664.842, eps=0.005042432
Epoch  63: f=   664.773, eps=0.005294554
Epoch  64: f=   664.707, eps=0.005559282
Epoch  65: f=   664.641, eps=0.005837246
```

```
Epoch  66: f=   664.578, eps=0.006129108
Epoch  67: f=   664.518, eps=0.006435563
Epoch  68: f=   664.467, eps=0.006757341
Epoch  69: f=   664.446, eps=0.007095208
Epoch  70: f=   664.544, eps=0.003547604
Epoch  71: f=   664.339, eps=0.003724984
Epoch  72: f=   664.278, eps=0.003911234
Epoch  73: f=   664.239, eps=0.004106795
Epoch  74: f=   664.206, eps=0.004312135
Epoch  75: f=   664.173, eps=0.004527742
Epoch  76: f=   664.139, eps=0.004754129
Epoch  77: f=   664.106, eps=0.004991835
Epoch  78: f=   664.072, eps=0.005241427
Epoch  79: f=   664.037, eps=0.005503499
Epoch  80: f=   664.002, eps=0.005778674
Epoch  81: f=   663.967, eps=0.006067607
Epoch  82: f=   663.936, eps=0.006370988
Epoch  83: f=   663.918, eps=0.006689537
Epoch  84: f=   663.948, eps=0.003344768
Epoch  85: f=   663.839, eps=0.003512007
Epoch  86: f=   663.807, eps=0.003687607
Epoch  87: f=   663.783, eps=0.003871988
Epoch  88: f=   663.760, eps=0.004065587
Epoch  89: f=   663.737, eps=0.004268866
Epoch  90: f=   663.713, eps=0.004482310
Epoch  91: f=   663.688, eps=0.004706425
Epoch  92: f=   663.661, eps=0.004941746
Epoch  93: f=   663.634, eps=0.005188834
Epoch  94: f=   663.606, eps=0.005448275
Epoch  95: f=   663.576, eps=0.005720689
Epoch  96: f=   663.546, eps=0.006006724
Epoch  97: f=   663.514, eps=0.006307060
Epoch  98: f=   663.482, eps=0.006622413
Epoch  99: f=   663.451, eps=0.006953533
Epoch 100: f=   663.427, eps=0.007301210
Epoch 101: f=   663.442, eps=0.003650605
Epoch 102: f=   663.371, eps=0.003833135
Epoch 103: f=   663.340, eps=0.004024792
Epoch 104: f=   663.316, eps=0.004226032
Epoch 105: f=   663.294, eps=0.004437333
Epoch 106: f=   663.271, eps=0.004659200
Epoch 107: f=   663.248, eps=0.004892160
Epoch 108: f=   663.223, eps=0.005136768
Epoch 109: f=   663.198, eps=0.005393606
Epoch 110: f=   663.172, eps=0.005663287
Epoch 111: f=   663.146, eps=0.005946451
Epoch 112: f=   663.121, eps=0.006243773
Epoch 113: f=   663.102, eps=0.006555962
Epoch 114: f=   663.108, eps=0.003277981
Epoch 115: f=   663.042, eps=0.003441880
Epoch 116: f=   663.019, eps=0.003613974
Epoch 117: f=   663.001, eps=0.003794673
Epoch 118: f=   662.982, eps=0.003984406
Epoch 119: f=   662.963, eps=0.004183627
Epoch 120: f=   662.943, eps=0.004392808
Epoch 121: f=   662.922, eps=0.004612449
Epoch 122: f=   662.900, eps=0.004843071
Epoch 123: f=   662.877, eps=0.005085225
Epoch 124: f=   662.853, eps=0.005339486
Epoch 125: f=   662.828, eps=0.005606460
Epoch 126: f=   662.802, eps=0.005886783
Epoch 127: f=   662.774, eps=0.006181122
Epoch 128: f=   662.745, eps=0.006490178
Epoch 129: f=   662.715, eps=0.006814687
Epoch 130: f=   662.685, eps=0.007155422
Epoch 131: f=   662.659, eps=0.007513193
Epoch 132: f=   662.656, eps=0.007888852
Epoch 133: f=   662.786, eps=0.003944426
Epoch 134: f=   662.631, eps=0.004141647
Epoch 135: f=   662.578, eps=0.004348730
Epoch 136: f=   662.545, eps=0.004566166
Epoch 137: f=   662.519, eps=0.004794475
Epoch 138: f=   662.497, eps=0.005034198
Epoch 139: f=   662.477, eps=0.005285908
Epoch 140: f=   662.462, eps=0.005550204
Epoch 141: f=   662.457, eps=0.005827714
Epoch 142: f=   662.476, eps=0.002913857
Epoch 143: f=   662.373, eps=0.003059550
Epoch 144: f=   662.355, eps=0.003212527
Epoch 145: f=   662.340, eps=0.003373154
Epoch 146: f=   662.325, eps=0.003541811
Epoch 147: f=   662.310, eps=0.003718902
Epoch 148: f=   662.293, eps=0.003904847
Epoch 149: f=   662.276, eps=0.004100089
Epoch 150: f=   662.257, eps=0.004305094
Epoch 151: f=   662.238, eps=0.004520348
Epoch 152: f=   662.218, eps=0.004746366
Epoch 153: f=   662.197, eps=0.004983684
Epoch 154: f=   662.175, eps=0.005232868
Epoch 155: f=   662.152, eps=0.005494512
Epoch 156: f=   662.128, eps=0.005769237
Epoch 157: f=   662.103, eps=0.006057699
```

```
Epoch 158: f=   662.076, eps=0.006360584
Epoch 159: f=   662.048, eps=0.006678613
Epoch 160: f=   662.019, eps=0.007012544
Epoch 161: f=   661.989, eps=0.007363171
Epoch 162: f=   661.957, eps=0.007731330
Epoch 163: f=   661.924, eps=0.008117896
Epoch 164: f=   661.890, eps=0.008523791
Epoch 165: f=   661.859, eps=0.008949981
Epoch 166: f=   661.868, eps=0.004474990
Epoch 167: f=   661.834, eps=0.004698740
Epoch 168: f=   661.809, eps=0.004933677
Epoch 169: f=   661.791, eps=0.005180361
Epoch 170: f=   661.780, eps=0.005439379
Epoch 171: f=   661.784, eps=0.002719689
Epoch 172: f=   661.698, eps=0.002855674
Epoch 173: f=   661.685, eps=0.002998458
Epoch 174: f=   661.672, eps=0.003148380
Epoch 175: f=   661.659, eps=0.003305799
Epoch 176: f=   661.645, eps=0.003471089
Epoch 177: f=   661.630, eps=0.003644644
Epoch 178: f=   661.615, eps=0.003826876
Epoch 179: f=   661.599, eps=0.004018220
Epoch 180: f=   661.582, eps=0.004219131
Epoch 181: f=   661.564, eps=0.004430087
Epoch 182: f=   661.546, eps=0.004651592
Epoch 183: f=   661.526, eps=0.004884171
Epoch 184: f=   661.506, eps=0.005128380
Epoch 185: f=   661.485, eps=0.005384799
Epoch 186: f=   661.462, eps=0.005654039
Epoch 187: f=   661.439, eps=0.005936741
Epoch 188: f=   661.414, eps=0.006233578
Epoch 189: f=   661.388, eps=0.006545257
Epoch 190: f=   661.361, eps=0.006872520
Epoch 191: f=   661.333, eps=0.007216146
Epoch 192: f=   661.303, eps=0.007576953
Epoch 193: f=   661.272, eps=0.007955801
Epoch 194: f=   661.240, eps=0.008353591
Epoch 195: f=   661.206, eps=0.008771270
Epoch 196: f=   661.170, eps=0.009209834
Epoch 197: f=   661.133, eps=0.009670325
Epoch 198: f=   661.097, eps=0.010153842
Epoch 199: f=   661.093, eps=0.010661534
Epoch 200: f=   661.463, eps=0.005330767
Epoch 201: f=   661.555, eps=0.002665383
Epoch 202: f=   660.978, eps=0.002798653
Epoch 203: f=   660.966, eps=0.002938585
Epoch 204: f=   660.955, eps=0.003085514
Epoch 205: f=   660.942, eps=0.003239790
Epoch 206: f=   660.929, eps=0.003401780
Epoch 207: f=   660.916, eps=0.003571869
Epoch 208: f=   660.902, eps=0.003750462
Epoch 209: f=   660.887, eps=0.003937985
Epoch 210: f=   660.871, eps=0.004134885
Epoch 211: f=   660.855, eps=0.004341629
Epoch 212: f=   660.837, eps=0.004558710
Epoch 213: f=   660.819, eps=0.004786646
Epoch 214: f=   660.801, eps=0.005025978
Epoch 215: f=   660.781, eps=0.005277277
Epoch 216: f=   660.760, eps=0.005541141
Epoch 217: f=   660.738, eps=0.005818198
Epoch 218: f=   660.715, eps=0.006109108
Epoch 219: f=   660.691, eps=0.006414563
Epoch 220: f=   660.666, eps=0.006735291
Epoch 221: f=   660.640, eps=0.007072056
Epoch 222: f=   660.612, eps=0.007425659
Epoch 223: f=   660.583, eps=0.007796941
Epoch 224: f=   660.553, eps=0.008186788
Epoch 225: f=   660.521, eps=0.008596128
Epoch 226: f=   660.488, eps=0.009025934
Epoch 227: f=   660.453, eps=0.009477231
Epoch 228: f=   660.417, eps=0.009951093
Epoch 229: f=   660.379, eps=0.010448647
Epoch 230: f=   660.344, eps=0.010971080
Epoch 231: f=   660.362, eps=0.005485540
Epoch 232: f=   660.377, eps=0.002742770
Epoch 233: f=   660.267, eps=0.002879908
Epoch 234: f=   660.254, eps=0.003023904
Epoch 235: f=   660.243, eps=0.003175099
Epoch 236: f=   660.231, eps=0.003333854
Epoch 237: f=   660.218, eps=0.003500547
Epoch 238: f=   660.205, eps=0.003675574
Epoch 239: f=   660.191, eps=0.003859353
Epoch 240: f=   660.176, eps=0.004052320
Epoch 241: f=   660.161, eps=0.004254936
Epoch 242: f=   660.145, eps=0.004467683
Epoch 243: f=   660.128, eps=0.004691067
Epoch 244: f=   660.111, eps=0.004925621
Epoch 245: f=   660.092, eps=0.005171902
Epoch 246: f=   660.073, eps=0.005430497
Epoch 247: f=   660.052, eps=0.005702022
Epoch 248: f=   660.031, eps=0.005987123
Epoch 249: f=   660.009, eps=0.006286479
```

```
Epoch 250: f=   659.985, eps=0.006600803
Epoch 251: f=   659.961, eps=0.006930843
Epoch 252: f=   659.935, eps=0.007277385
Epoch 253: f=   659.908, eps=0.007641254
Epoch 254: f=   659.880, eps=0.008023317
Epoch 255: f=   659.850, eps=0.008424483
Epoch 256: f=   659.819, eps=0.008845707
Epoch 257: f=   659.787, eps=0.009287992
Epoch 258: f=   659.754, eps=0.009752392
Epoch 259: f=   659.737, eps=0.010240012
Epoch 260: f=   659.888, eps=0.005120006
Epoch 261: f=   659.906, eps=0.002560003
Epoch 262: f=   659.651, eps=0.002688003
Epoch 263: f=   659.641, eps=0.002822403
Epoch 264: f=   659.631, eps=0.002963523
Epoch 265: f=   659.620, eps=0.003111700
Epoch 266: f=   659.609, eps=0.003267285
Epoch 267: f=   659.597, eps=0.003430649
Epoch 268: f=   659.585, eps=0.003602181
Epoch 269: f=   659.572, eps=0.003782290
Epoch 270: f=   659.558, eps=0.003971405
Epoch 271: f=   659.543, eps=0.004169975
Epoch 272: f=   659.528, eps=0.004378474
Epoch 273: f=   659.513, eps=0.004597397
Epoch 274: f=   659.496, eps=0.004827267
Epoch 275: f=   659.479, eps=0.005068631
Epoch 276: f=   659.460, eps=0.005322062
Epoch 277: f=   659.441, eps=0.005588165
Epoch 278: f=   659.421, eps=0.005867574
Epoch 279: f=   659.400, eps=0.006160952
Epoch 280: f=   659.378, eps=0.006469000
Epoch 281: f=   659.355, eps=0.006792450
Epoch 282: f=   659.331, eps=0.007132072
Epoch 283: f=   659.305, eps=0.007488676
Epoch 284: f=   659.279, eps=0.007863110
Epoch 285: f=   659.251, eps=0.008256265
Epoch 286: f=   659.222, eps=0.008669078
Epoch 287: f=   659.191, eps=0.009102532
Epoch 288: f=   659.159, eps=0.009557659
Epoch 289: f=   659.125, eps=0.010035542
Epoch 290: f=   659.090, eps=0.010537319
Epoch 291: f=   659.053, eps=0.011064185
Epoch 292: f=   659.016, eps=0.011617394
Epoch 293: f=   658.992, eps=0.012198264
Epoch 294: f=   659.226, eps=0.006099132
Epoch 295: f=   659.526, eps=0.003049566
Epoch 296: f=   658.916, eps=0.003202044
Epoch 297: f=   658.891, eps=0.003362147
Epoch 298: f=   658.878, eps=0.003530254
Epoch 299: f=   658.865, eps=0.003706767
Epoch 300: f=   658.852, eps=0.003892105
Epoch 301: f=   658.839, eps=0.004086710
Epoch 302: f=   658.825, eps=0.004291046
Epoch 303: f=   658.810, eps=0.004505598
Epoch 304: f=   658.795, eps=0.004730878
Epoch 305: f=   658.778, eps=0.004967422
Epoch 306: f=   658.761, eps=0.005215793
Epoch 307: f=   658.743, eps=0.005476582
Epoch 308: f=   658.725, eps=0.005750412
Epoch 309: f=   658.705, eps=0.006037932
Epoch 310: f=   658.684, eps=0.006339829
Epoch 311: f=   658.663, eps=0.006656820
Epoch 312: f=   658.640, eps=0.006989661
Epoch 313: f=   658.617, eps=0.007339144
Epoch 314: f=   658.593, eps=0.007706101
Epoch 315: f=   658.573, eps=0.008091406
Epoch 316: f=   658.582, eps=0.004045703
Epoch 317: f=   658.544, eps=0.004247988
Epoch 318: f=   658.521, eps=0.004460388
Epoch 319: f=   658.503, eps=0.004683407
Epoch 320: f=   658.486, eps=0.004917578
Epoch 321: f=   658.470, eps=0.005163456
Epoch 322: f=   658.455, eps=0.005421629
Epoch 323: f=   658.443, eps=0.005692711
Epoch 324: f=   658.436, eps=0.005977346
Epoch 325: f=   658.450, eps=0.002988673
Epoch 326: f=   658.381, eps=0.003138107
Epoch 327: f=   658.368, eps=0.003295012
Epoch 328: f=   658.356, eps=0.003459763
Epoch 329: f=   658.345, eps=0.003632751
Epoch 330: f=   658.333, eps=0.003814388
Epoch 331: f=   658.320, eps=0.004005108
Epoch 332: f=   658.307, eps=0.004205363
Epoch 333: f=   658.293, eps=0.004415631
Epoch 334: f=   658.278, eps=0.004636413
Epoch 335: f=   658.263, eps=0.004868234
Epoch 336: f=   658.247, eps=0.005111645
Epoch 337: f=   658.230, eps=0.005367228
Epoch 338: f=   658.212, eps=0.005635589
Epoch 339: f=   658.193, eps=0.005917368
Epoch 340: f=   658.174, eps=0.006213237
Epoch 341: f=   658.153, eps=0.006523899
```

```
Epoch 342: f=   658.132, eps=0.006850094
Epoch 343: f=   658.109, eps=0.007192598
Epoch 344: f=   658.086, eps=0.007552228
Epoch 345: f=   658.061, eps=0.007929840
Epoch 346: f=   658.036, eps=0.008326332
Epoch 347: f=   658.017, eps=0.008742648
Epoch 348: f=   658.040, eps=0.004371324
Epoch 349: f=   658.004, eps=0.004589890
Epoch 350: f=   657.981, eps=0.004819385
Epoch 351: f=   657.965, eps=0.005060354
Epoch 352: f=   657.954, eps=0.005313372
Epoch 353: f=   657.953, eps=0.005579040
Epoch 354: f=   657.969, eps=0.002789520
Epoch 355: f=   657.876, eps=0.002928996
Epoch 356: f=   657.864, eps=0.003075446
Epoch 357: f=   657.854, eps=0.003229218
Epoch 358: f=   657.844, eps=0.003390679
Epoch 359: f=   657.833, eps=0.003560213
Epoch 360: f=   657.821, eps=0.003738224
Epoch 361: f=   657.809, eps=0.003925135
Epoch 362: f=   657.797, eps=0.004121392
Epoch 363: f=   657.783, eps=0.004327461
Epoch 364: f=   657.770, eps=0.004543834
Epoch 365: f=   657.755, eps=0.004771026
Epoch 366: f=   657.740, eps=0.005009577
Epoch 367: f=   657.724, eps=0.005260056
Epoch 368: f=   657.707, eps=0.005523059
Epoch 369: f=   657.689, eps=0.005799212
Epoch 370: f=   657.671, eps=0.006089173
Epoch 371: f=   657.651, eps=0.006393631
Epoch 372: f=   657.631, eps=0.006713313
Epoch 373: f=   657.609, eps=0.007048978
Epoch 374: f=   657.587, eps=0.007401427
Epoch 375: f=   657.564, eps=0.007771499
Epoch 376: f=   657.539, eps=0.008160074
Epoch 377: f=   657.513, eps=0.008568077
Epoch 378: f=   657.486, eps=0.008996481
Epoch 379: f=   657.460, eps=0.009446305
Epoch 380: f=   657.445, eps=0.009918621
Epoch 381: f=   657.554, eps=0.004959310
Epoch 382: f=   657.540, eps=0.005207276
Epoch 383: f=   657.567, eps=0.002603638
Epoch 384: f=   657.357, eps=0.002733820
Epoch 385: f=   657.348, eps=0.002870511
Epoch 386: f=   657.339, eps=0.003014036
Epoch 387: f=   657.330, eps=0.003164738
Epoch 388: f=   657.320, eps=0.003322975
Epoch 389: f=   657.310, eps=0.003489124
Epoch 390: f=   657.299, eps=0.003663580
Epoch 391: f=   657.287, eps=0.003846759
Epoch 392: f=   657.275, eps=0.004039097
Epoch 393: f=   657.263, eps=0.004241052
Epoch 394: f=   657.250, eps=0.004453104
Epoch 395: f=   657.236, eps=0.004675760
Epoch 396: f=   657.221, eps=0.004909548
Epoch 397: f=   657.206, eps=0.005155025
Epoch 398: f=   657.190, eps=0.005412776
Epoch 399: f=   657.173, eps=0.005683415
Epoch 400: f=   657.156, eps=0.005967586
Epoch 401: f=   657.138, eps=0.006265965
Epoch 402: f=   657.118, eps=0.006579263
Epoch 403: f=   657.098, eps=0.006908226
Epoch 404: f=   657.077, eps=0.007253638
Epoch 405: f=   657.054, eps=0.007616320
Epoch 406: f=   657.031, eps=0.007997136
Epoch 407: f=   657.007, eps=0.008396992
Epoch 408: f=   656.981, eps=0.008816842
Epoch 409: f=   656.954, eps=0.009257684
Epoch 410: f=   656.926, eps=0.009720568
Epoch 411: f=   656.896, eps=0.010206597
Epoch 412: f=   656.866, eps=0.010716927
Epoch 413: f=   656.838, eps=0.011252773
Epoch 414: f=   656.871, eps=0.005626387
Epoch 415: f=   656.908, eps=0.002813193
Epoch 416: f=   656.776, eps=0.002953853
Epoch 417: f=   656.765, eps=0.003101546
Epoch 418: f=   656.755, eps=0.003256623
Epoch 419: f=   656.745, eps=0.003419454
Epoch 420: f=   656.735, eps=0.003590427
Epoch 421: f=   656.724, eps=0.003769948
Epoch 422: f=   656.713, eps=0.003958445
Epoch 423: f=   656.701, eps=0.004156368
Epoch 424: f=   656.689, eps=0.004364186
Epoch 425: f=   656.676, eps=0.004582395
Epoch 426: f=   656.662, eps=0.004811515
Epoch 427: f=   656.648, eps=0.005052091
Epoch 428: f=   656.632, eps=0.005304695
Epoch 429: f=   656.617, eps=0.005569930
Epoch 430: f=   656.600, eps=0.005848427
Epoch 431: f=   656.583, eps=0.006140848
Epoch 432: f=   656.564, eps=0.006447890
Epoch 433: f=   656.545, eps=0.006770285
```

```
Epoch 434: f=   656.525, eps=0.007108799
Epoch 435: f=   656.504, eps=0.007464239
Epoch 436: f=   656.482, eps=0.007837451
Epoch 437: f=   656.459, eps=0.008229324
Epoch 438: f=   656.435, eps=0.008640790
Epoch 439: f=   656.410, eps=0.009072829
Epoch 440: f=   656.388, eps=0.009526471
Epoch 441: f=   656.406, eps=0.004763235
Epoch 442: f=   656.387, eps=0.005001397
Epoch 443: f=   656.379, eps=0.005251467
Epoch 444: f=   656.381, eps=0.002625734
Epoch 445: f=   656.303, eps=0.002757020
Epoch 446: f=   656.295, eps=0.002894871
Epoch 447: f=   656.286, eps=0.003039615
Epoch 448: f=   656.277, eps=0.003191596
Epoch 449: f=   656.268, eps=0.003351175
Epoch 450: f=   656.258, eps=0.003518734
Epoch 451: f=   656.248, eps=0.003694671
Epoch 452: f=   656.237, eps=0.003879404
Epoch 453: f=   656.226, eps=0.004073375
Epoch 454: f=   656.214, eps=0.004277043
Epoch 455: f=   656.202, eps=0.004490895
Epoch 456: f=   656.189, eps=0.004715440
Epoch 457: f=   656.175, eps=0.004951212
Epoch 458: f=   656.161, eps=0.005198773
Epoch 459: f=   656.145, eps=0.005458711
Epoch 460: f=   656.130, eps=0.005731647
Epoch 461: f=   656.113, eps=0.006018229
Epoch 462: f=   656.096, eps=0.006319141
Epoch 463: f=   656.077, eps=0.006635098
Epoch 464: f=   656.058, eps=0.006966853
Epoch 465: f=   656.038, eps=0.007315195
Epoch 466: f=   656.017, eps=0.007680955
Epoch 467: f=   655.995, eps=0.008065003
Epoch 468: f=   655.972, eps=0.008468253
Epoch 469: f=   655.948, eps=0.008891666
Epoch 470: f=   655.923, eps=0.009336249
Epoch 471: f=   655.896, eps=0.009803061
Epoch 472: f=   655.868, eps=0.010293215
Epoch 473: f=   655.841, eps=0.010807875
Epoch 474: f=   655.835, eps=0.011348269
Epoch 475: f=   656.135, eps=0.005674135
Epoch 476: f=   656.301, eps=0.002837067
Epoch 477: f=   655.760, eps=0.002978921
Epoch 478: f=   655.744, eps=0.003127867
Epoch 479: f=   655.735, eps=0.003284260
Epoch 480: f=   655.725, eps=0.003448473
Epoch 481: f=   655.716, eps=0.003620897
Epoch 482: f=   655.705, eps=0.003801941
Epoch 483: f=   655.695, eps=0.003992039
Epoch 484: f=   655.684, eps=0.004191640
Epoch 485: f=   655.672, eps=0.004401222
Epoch 486: f=   655.659, eps=0.004621284
Epoch 487: f=   655.646, eps=0.004852348
Epoch 488: f=   655.633, eps=0.005094965
Epoch 489: f=   655.619, eps=0.005349713
Epoch 490: f=   655.604, eps=0.005617199
Epoch 491: f=   655.588, eps=0.005898059
Epoch 492: f=   655.571, eps=0.006192962
Epoch 493: f=   655.554, eps=0.006502610
Epoch 494: f=   655.536, eps=0.006827741
Epoch 495: f=   655.517, eps=0.007169128
Epoch 496: f=   655.497, eps=0.007527584
Epoch 497: f=   655.476, eps=0.007903963
Epoch 498: f=   655.454, eps=0.008299161
Epoch 499: f=   655.432, eps=0.008714119
Result after 500 epochs: f=655.4134964699465
```

In [25]:

```python
# look at how gradient descent made progess
# YOUR CODE HERE
nextplot()
plt.plot(range(501),vz_gd)
plt.show()
```

2d Stochastic gradient descent

In [26]:

```python
def sgdepoch(y, X, w, eps):
    """Run one SGD epoch and return the updated weight vector. """
    # Run N stochastic gradient steps (without replacement). Do not rescale each
    # step by factor N (i.e., proceed differently than in the lecture slides).
    # YOUR CODE HERE
    # np.random.rand(2022)
    # np.random.shuffle(X)
    # np.random.shuffle(y)
    index_list = list(range(X.shape[0]))
    np.random.shuffle(index_list)
    for i in index_list :
```

```
        gradient = dl(y[i],X[i],w)
        w += eps * gradient
        return w
```

In [27]:

```
# when you run this multiple times, with 50% probability you should get the
# following result (there is one other result which is very close):
# array([ -3.43689655e+02,  -1.71161311e+02,  -5.71093536e+02,
#         -5.16478220e+01,   4.66294348e+02,  -3.71589878e+02,
#          5.21493183e+02,   1.25699230e+03,   8.33804130e+02,
#          5.63185399e+02,   1.32761302e+03,  -2.64104011e+02,
#          7.10693307e+02,  -1.75497331e+02,  -1.94174427e+02,
#          1.11641507e+02,  -3.30817509e+02,  -3.46754913e+02,
#          8.48722111e+02,  -1.89136304e+02,  -4.25693844e+02,
#         -1.23084189e+02,  -2.95894797e+02,  -2.35789333e+02,
#         -3.38695243e+02,  -3.05642830e+02,  -2.28975383e+02,
#         -2.38075137e+02,  -1.66702530e+02,  -2.27341599e+02,
#         -1.77575620e+02,  -1.49093855e+02,  -1.70028859e+02,
#         -1.50243833e+02,  -1.82986008e+02,  -2.41143708e+02,
#         -3.31047159e+02,  -5.79991185e+01,  -1.98477863e+02,
#         -1.91264948e+02,  -1.17371919e+02,  -1.66953779e+02,
#         -2.01472565e+02,  -1.23330949e+02,  -3.00857740e+02,
#         -1.95853348e+02,  -7.44868073e+01,  -1.11172370e+02,
#         -1.57618226e+02,  -1.25729512e+00,  -1.45536466e+02,
#         -1.43362438e+02,  -3.00429708e+02,  -9.84391082e+01,
#         -4.54152047e+01,  -5.26492232e+01,  -1.45175427e+02])
sgdepoch(y[1:3], Xz[1:3, :], np.linspace(-5, 5, D), 1000)
```

Out[27]:

```
array([-3.43689655e+02, -1.71161311e+02, -5.71093536e+02, -5.16478220e+01,
        4.66294348e+02, -3.71589878e+02,  5.21493183e+02,  1.25699230e+03,
        8.33804130e+02,  5.63185399e+02,  1.32761302e+03, -2.64104011e+02,
        7.10693307e+02, -1.75497331e+02, -1.94174427e+02,  1.11641507e+02,
       -3.30817509e+02, -3.46754913e+02,  8.48722111e+02, -1.89136304e+02,
       -4.25693844e+02, -1.23084189e+02, -2.95894797e+02, -2.35789333e+02,
       -3.38695243e+02, -3.05642830e+02, -2.28975383e+02, -2.38075137e+02,
       -1.66702530e+02, -2.27341599e+02, -1.77575620e+02, -1.49093855e+02,
       -1.70028859e+02, -1.50243833e+02, -1.82986008e+02, -2.41143708e+02,
       -3.31047159e+02, -5.79991185e+01, -1.98477863e+02, -1.91264948e+02,
       -1.17371919e+02, -1.66953779e+02, -2.01472565e+02, -1.23330949e+02,
       -3.00857740e+02, -1.95853348e+02, -7.44868073e+01, -1.11172370e+02,
       -1.57618226e+02, -1.25729512e+00, -1.45536466e+02, -1.43362438e+02,
       -3.00429708e+02, -9.84391082e+01, -4.54152047e+01, -5.26492232e+01,
       -1.45175427e+02])
```

In [28]:

```
# define the objective and update function for one gradient-descent epoch for
# fitting an MLE estimate of logistic regression with stochastic gradient descent
# (should return a tuple of two functions; see optimize)
def sgd(y, X):
    def objective(w):
        # YOUR CODE HERE
        p_i0 = logsigma(-np.dot(X,w))
        p_i1 = logsigma(np.dot(X,w))
        objective = -np.sum(y*p_i1 + (1-y)*p_i0)
        return objective

    def update(w, eps):
        return sgdepoch(y, X, w, eps)

    return (objective, update)
```

In [29]:

```
# with 50% probability, you should get:
# [40.864973045695081,
#  array([ -3.43689655e+02,  -1.71161311e+02,  -5.71093536e+02,
#          -5.16478220e+01,   4.66294348e+02,  -3.71589878e+02,
#           5.21493183e+02,   1.25699230e+03,   8.33804130e+02,
#           5.63185399e+02,   1.32761302e+03,  -2.64104011e+02,
#           7.10693307e+02,  -1.75497331e+02,  -1.94174427e+02,
#           1.11641507e+02,  -3.30817509e+02,  -3.46754913e+02,
#           8.48722111e+02,  -1.89136304e+02,  -4.25693844e+02,
#          -1.23084189e+02,  -2.95894797e+02,  -2.35789333e+02,
#          -3.38695243e+02,  -3.05642830e+02,  -2.28975383e+02,
#          -2.38075137e+02,  -1.66702530e+02,  -2.27341599e+02,
#          -1.77575620e+02,  -1.49093855e+02,  -1.70028859e+02,
#          -1.50243833e+02,  -1.82986008e+02,  -2.41143708e+02,
#          -3.31047159e+02,  -5.79991185e+01,  -1.98477863e+02,
#          -1.91264948e+02,  -1.17371919e+02,  -1.66953779e+02,
#          -2.01472565e+02,  -1.23330949e+02,  -3.00857740e+02,
#          -1.95853348e+02,  -7.44868073e+01,  -1.11172370e+02,
#          -1.57618226e+02,  -1.25729512e+00,  -1.45536466e+02,
#          -1.43362438e+02,  -3.00429708e+02,  -9.84391082e+01,
#          -4.54152047e+01,  -5.26492232e+01,  -1.45175427e+02])]
f, update = sgd(y[1:3], Xz[1:3, :])
[f(np.linspace(-5, 5, D)), update(np.linspace(-5, 5, D), 1000)]
```

Out[29]:

```
[40.864973045695095,
 array([-3.43689655e+02, -1.71161311e+02, -5.71093536e+02, -5.16478220e+01,
         4.66294348e+02, -3.71589878e+02,  5.21493183e+02,  1.25699230e+03,
         8.33804130e+02,  5.63185399e+02,  1.32761302e+03, -2.64104011e+02,
         7.10693307e+02, -1.75497331e+02, -1.94174427e+02,  1.11641507e+02,
```

```
        -3.30817509e+02, -3.46754913e+02,  8.48722111e+02, -1.89136304e+02,
        -4.25693844e+02, -1.23084189e+02, -2.95894797e+02, -2.35789333e+02,
        -3.38695243e+02, -3.05642830e+02, -2.28975383e+02, -2.38075137e+02,
        -1.66702530e+02, -2.27341599e+02, -1.77575620e+02, -1.49093855e+02,
        -1.70028859e+02, -1.50243833e+02, -1.82986008e+02, -2.41143708e+02,
        -3.31047159e+02, -5.79991185e+01, -1.98477863e+02, -1.91264948e+02,
        -1.17371919e+02, -1.66953779e+02, -2.01472565e+02, -1.23330949e+02,
        -3.00857740e+02, -1.95853348e+02, -7.44868073e+01, -1.11172370e+02,
        -1.57618226e+02, -1.25729512e+00, -1.45536466e+02, -1.43362438e+02,
        -3.00429708e+02, -9.84391082e+01, -4.54152047e+01, -5.26492232e+01,
        -1.45175427e+02])]
```

In [30]:

```
# you can run stochastic gradient descent!
wz_sgd, vz_sgd, ez_sgd = optimize(sgd(y, Xz), w0, nepochs=500)
```

```
Epoch   0: f=   655.413, eps=0.010000000
Epoch   1: f=   655.413, eps=0.010500000
Epoch   2: f=   655.413, eps=0.011025000
Epoch   3: f=   655.413, eps=0.005512500
Epoch   4: f=   655.414, eps=0.002756250
Epoch   5: f=   655.439, eps=0.001378125
Epoch   6: f=   655.439, eps=0.000689063
Epoch   7: f=   655.439, eps=0.000344531
Epoch   8: f=   655.439, eps=0.000172266
Epoch   9: f=   655.439, eps=0.000086133
Epoch  10: f=   655.439, eps=0.000043066
Epoch  11: f=   655.439, eps=0.000045220
Epoch  12: f=   655.439, eps=0.000022610
Epoch  13: f=   655.439, eps=0.000023740
Epoch  14: f=   655.439, eps=0.000011870
Epoch  15: f=   655.439, eps=0.000005935
Epoch  16: f=   655.439, eps=0.000006232
Epoch  17: f=   655.439, eps=0.000006543
Epoch  18: f=   655.439, eps=0.000003272
Epoch  19: f=   655.439, eps=0.000001636
Epoch  20: f=   655.439, eps=0.000000818
Epoch  21: f=   655.439, eps=0.000000409
Epoch  22: f=   655.439, eps=0.000000429
Epoch  23: f=   655.439, eps=0.000000215
Epoch  24: f=   655.439, eps=0.000000107
Epoch  25: f=   655.439, eps=0.000000054
Epoch  26: f=   655.439, eps=0.000000027
Epoch  27: f=   655.439, eps=0.000000013
Epoch  28: f=   655.439, eps=0.000000014
Epoch  29: f=   655.439, eps=0.000000007
Epoch  30: f=   655.439, eps=0.000000004
Epoch  31: f=   655.439, eps=0.000000002
Epoch  32: f=   655.439, eps=0.000000001
Epoch  33: f=   655.439, eps=0.000000000
Epoch  34: f=   655.439, eps=0.000000000
Epoch  35: f=   655.439, eps=0.000000000
Epoch  36: f=   655.439, eps=0.000000000
Epoch  37: f=   655.439, eps=0.000000000
Epoch  38: f=   655.439, eps=0.000000000
Epoch  39: f=   655.439, eps=0.000000000
Epoch  40: f=   655.439, eps=0.000000000
Epoch  41: f=   655.439, eps=0.000000000
Epoch  42: f=   655.439, eps=0.000000000
Epoch  43: f=   655.439, eps=0.000000000
Epoch  44: f=   655.439, eps=0.000000000
Epoch  45: f=   655.439, eps=0.000000000
Epoch  46: f=   655.439, eps=0.000000000
Epoch  47: f=   655.439, eps=0.000000000
Epoch  48: f=   655.439, eps=0.000000000
Epoch  49: f=   655.439, eps=0.000000000
Epoch  50: f=   655.439, eps=0.000000000
Epoch  51: f=   655.439, eps=0.000000000
Epoch  52: f=   655.439, eps=0.000000000
Epoch  53: f=   655.439, eps=0.000000000
Epoch  54: f=   655.439, eps=0.000000000
Epoch  55: f=   655.439, eps=0.000000000
Epoch  56: f=   655.439, eps=0.000000000
Epoch  57: f=   655.439, eps=0.000000000
Epoch  58: f=   655.439, eps=0.000000000
Epoch  59: f=   655.439, eps=0.000000000
Epoch  60: f=   655.439, eps=0.000000000
Epoch  61: f=   655.439, eps=0.000000000
Epoch  62: f=   655.439, eps=0.000000000
Epoch  63: f=   655.439, eps=0.000000000
Epoch  64: f=   655.439, eps=0.000000000
Epoch  65: f=   655.439, eps=0.000000000
Epoch  66: f=   655.439, eps=0.000000000
Epoch  67: f=   655.439, eps=0.000000000
Epoch  68: f=   655.439, eps=0.000000000
Epoch  69: f=   655.439, eps=0.000000000
Epoch  70: f=   655.439, eps=0.000000000
Epoch  71: f=   655.439, eps=0.000000000
Epoch  72: f=   655.439, eps=0.000000000
Epoch  73: f=   655.439, eps=0.000000000
Epoch  74: f=   655.439, eps=0.000000000
Epoch  75: f=   655.439, eps=0.000000000
```

```
Epoch  76: f=   655.439, eps=0.000000000
Epoch  77: f=   655.439, eps=0.000000000
Epoch  78: f=   655.439, eps=0.000000000
Epoch  79: f=   655.439, eps=0.000000000
Epoch  80: f=   655.439, eps=0.000000000
Epoch  81: f=   655.439, eps=0.000000000
Epoch  82: f=   655.439, eps=0.000000000
Epoch  83: f=   655.439, eps=0.000000000
Epoch  84: f=   655.439, eps=0.000000000
Epoch  85: f=   655.439, eps=0.000000000
Epoch  86: f=   655.439, eps=0.000000000
Epoch  87: f=   655.439, eps=0.000000000
Epoch  88: f=   655.439, eps=0.000000000
Epoch  89: f=   655.439, eps=0.000000000
Epoch  90: f=   655.439, eps=0.000000000
Epoch  91: f=   655.439, eps=0.000000000
Epoch  92: f=   655.439, eps=0.000000000
Epoch  93: f=   655.439, eps=0.000000000
Epoch  94: f=   655.439, eps=0.000000000
Epoch  95: f=   655.439, eps=0.000000000
Epoch  96: f=   655.439, eps=0.000000000
Epoch  97: f=   655.439, eps=0.000000000
Epoch  98: f=   655.439, eps=0.000000000
Epoch  99: f=   655.439, eps=0.000000000
Epoch 100: f=   655.439, eps=0.000000000
Epoch 101: f=   655.439, eps=0.000000000
Epoch 102: f=   655.439, eps=0.000000000
Epoch 103: f=   655.439, eps=0.000000000
Epoch 104: f=   655.439, eps=0.000000000
Epoch 105: f=   655.439, eps=0.000000000
Epoch 106: f=   655.439, eps=0.000000000
Epoch 107: f=   655.439, eps=0.000000000
Epoch 108: f=   655.439, eps=0.000000000
Epoch 109: f=   655.439, eps=0.000000000
Epoch 110: f=   655.439, eps=0.000000000
Epoch 111: f=   655.439, eps=0.000000000
Epoch 112: f=   655.439, eps=0.000000000
Epoch 113: f=   655.439, eps=0.000000000
Epoch 114: f=   655.439, eps=0.000000000
Epoch 115: f=   655.439, eps=0.000000000
Epoch 116: f=   655.439, eps=0.000000000
Epoch 117: f=   655.439, eps=0.000000000
Epoch 118: f=   655.439, eps=0.000000000
Epoch 119: f=   655.439, eps=0.000000000
Epoch 120: f=   655.439, eps=0.000000000
Epoch 121: f=   655.439, eps=0.000000000
Epoch 122: f=   655.439, eps=0.000000000
Epoch 123: f=   655.439, eps=0.000000000
Epoch 124: f=   655.439, eps=0.000000000
Epoch 125: f=   655.439, eps=0.000000000
Epoch 126: f=   655.439, eps=0.000000000
Epoch 127: f=   655.439, eps=0.000000000
Epoch 128: f=   655.439, eps=0.000000000
Epoch 129: f=   655.439, eps=0.000000000
Epoch 130: f=   655.439, eps=0.000000000
Epoch 131: f=   655.439, eps=0.000000000
Epoch 132: f=   655.439, eps=0.000000000
Epoch 133: f=   655.439, eps=0.000000000
Epoch 134: f=   655.439, eps=0.000000000
Epoch 135: f=   655.439, eps=0.000000000
Epoch 136: f=   655.439, eps=0.000000000
Epoch 137: f=   655.439, eps=0.000000000
Epoch 138: f=   655.439, eps=0.000000000
Epoch 139: f=   655.439, eps=0.000000000
Epoch 140: f=   655.439, eps=0.000000000
Epoch 141: f=   655.439, eps=0.000000000
Epoch 142: f=   655.439, eps=0.000000000
Epoch 143: f=   655.439, eps=0.000000000
Epoch 144: f=   655.439, eps=0.000000000
Epoch 145: f=   655.439, eps=0.000000000
Epoch 146: f=   655.439, eps=0.000000000
Epoch 147: f=   655.439, eps=0.000000000
Epoch 148: f=   655.439, eps=0.000000000
Epoch 149: f=   655.439, eps=0.000000000
Epoch 150: f=   655.439, eps=0.000000000
Epoch 151: f=   655.439, eps=0.000000000
Epoch 152: f=   655.439, eps=0.000000000
Epoch 153: f=   655.439, eps=0.000000000
Epoch 154: f=   655.439, eps=0.000000000
Epoch 155: f=   655.439, eps=0.000000000
Epoch 156: f=   655.439, eps=0.000000000
Epoch 157: f=   655.439, eps=0.000000000
Epoch 158: f=   655.439, eps=0.000000000
Epoch 159: f=   655.439, eps=0.000000000
Epoch 160: f=   655.439, eps=0.000000000
Epoch 161: f=   655.439, eps=0.000000000
Epoch 162: f=   655.439, eps=0.000000000
Epoch 163: f=   655.439, eps=0.000000000
Epoch 164: f=   655.439, eps=0.000000000
Epoch 165: f=   655.439, eps=0.000000000
Epoch 166: f=   655.439, eps=0.000000000
Epoch 167: f=   655.439, eps=0.000000000
```

```
Epoch 168: f=   655.439, eps=0.000000000
Epoch 169: f=   655.439, eps=0.000000000
Epoch 170: f=   655.439, eps=0.000000000
Epoch 171: f=   655.439, eps=0.000000000
Epoch 172: f=   655.439, eps=0.000000000
Epoch 173: f=   655.439, eps=0.000000000
Epoch 174: f=   655.439, eps=0.000000000
Epoch 175: f=   655.439, eps=0.000000000
Epoch 176: f=   655.439, eps=0.000000000
Epoch 177: f=   655.439, eps=0.000000000
Epoch 178: f=   655.439, eps=0.000000000
Epoch 179: f=   655.439, eps=0.000000000
Epoch 180: f=   655.439, eps=0.000000000
Epoch 181: f=   655.439, eps=0.000000000
Epoch 182: f=   655.439, eps=0.000000000
Epoch 183: f=   655.439, eps=0.000000000
Epoch 184: f=   655.439, eps=0.000000000
Epoch 185: f=   655.439, eps=0.000000000
Epoch 186: f=   655.439, eps=0.000000000
Epoch 187: f=   655.439, eps=0.000000000
Epoch 188: f=   655.439, eps=0.000000000
Epoch 189: f=   655.439, eps=0.000000000
Epoch 190: f=   655.439, eps=0.000000000
Epoch 191: f=   655.439, eps=0.000000000
Epoch 192: f=   655.439, eps=0.000000000
Epoch 193: f=   655.439, eps=0.000000000
Epoch 194: f=   655.439, eps=0.000000000
Epoch 195: f=   655.439, eps=0.000000000
Epoch 196: f=   655.439, eps=0.000000000
Epoch 197: f=   655.439, eps=0.000000000
Epoch 198: f=   655.439, eps=0.000000000
Epoch 199: f=   655.439, eps=0.000000000
Epoch 200: f=   655.439, eps=0.000000000
Epoch 201: f=   655.439, eps=0.000000000
Epoch 202: f=   655.439, eps=0.000000000
Epoch 203: f=   655.439, eps=0.000000000
Epoch 204: f=   655.439, eps=0.000000000
Epoch 205: f=   655.439, eps=0.000000000
Epoch 206: f=   655.439, eps=0.000000000
Epoch 207: f=   655.439, eps=0.000000000
Epoch 208: f=   655.439, eps=0.000000000
Epoch 209: f=   655.439, eps=0.000000000
Epoch 210: f=   655.439, eps=0.000000000
Epoch 211: f=   655.439, eps=0.000000000
Epoch 212: f=   655.439, eps=0.000000000
Epoch 213: f=   655.439, eps=0.000000000
Epoch 214: f=   655.439, eps=0.000000000
Epoch 215: f=   655.439, eps=0.000000000
Epoch 216: f=   655.439, eps=0.000000000
Epoch 217: f=   655.439, eps=0.000000000
Epoch 218: f=   655.439, eps=0.000000000
Epoch 219: f=   655.439, eps=0.000000000
Epoch 220: f=   655.439, eps=0.000000000
Epoch 221: f=   655.439, eps=0.000000000
Epoch 222: f=   655.439, eps=0.000000000
Epoch 223: f=   655.439, eps=0.000000000
Epoch 224: f=   655.439, eps=0.000000000
Epoch 225: f=   655.439, eps=0.000000000
Epoch 226: f=   655.439, eps=0.000000000
Epoch 227: f=   655.439, eps=0.000000000
Epoch 228: f=   655.439, eps=0.000000000
Epoch 229: f=   655.439, eps=0.000000000
Epoch 230: f=   655.439, eps=0.000000000
Epoch 231: f=   655.439, eps=0.000000000
Epoch 232: f=   655.439, eps=0.000000000
Epoch 233: f=   655.439, eps=0.000000000
Epoch 234: f=   655.439, eps=0.000000000
Epoch 235: f=   655.439, eps=0.000000000
Epoch 236: f=   655.439, eps=0.000000000
Epoch 237: f=   655.439, eps=0.000000000
Epoch 238: f=   655.439, eps=0.000000000
Epoch 239: f=   655.439, eps=0.000000000
Epoch 240: f=   655.439, eps=0.000000000
Epoch 241: f=   655.439, eps=0.000000000
Epoch 242: f=   655.439, eps=0.000000000
Epoch 243: f=   655.439, eps=0.000000000
Epoch 244: f=   655.439, eps=0.000000000
Epoch 245: f=   655.439, eps=0.000000000
Epoch 246: f=   655.439, eps=0.000000000
Epoch 247: f=   655.439, eps=0.000000000
Epoch 248: f=   655.439, eps=0.000000000
Epoch 249: f=   655.439, eps=0.000000000
Epoch 250: f=   655.439, eps=0.000000000
Epoch 251: f=   655.439, eps=0.000000000
Epoch 252: f=   655.439, eps=0.000000000
Epoch 253: f=   655.439, eps=0.000000000
Epoch 254: f=   655.439, eps=0.000000000
Epoch 255: f=   655.439, eps=0.000000000
Epoch 256: f=   655.439, eps=0.000000000
Epoch 257: f=   655.439, eps=0.000000000
Epoch 258: f=   655.439, eps=0.000000000
Epoch 259: f=   655.439, eps=0.000000000
```

```
Epoch 260: f=   655.439, eps=0.000000000
Epoch 261: f=   655.439, eps=0.000000000
Epoch 262: f=   655.439, eps=0.000000000
Epoch 263: f=   655.439, eps=0.000000000
Epoch 264: f=   655.439, eps=0.000000000
Epoch 265: f=   655.439, eps=0.000000000
Epoch 266: f=   655.439, eps=0.000000000
Epoch 267: f=   655.439, eps=0.000000000
Epoch 268: f=   655.439, eps=0.000000000
Epoch 269: f=   655.439, eps=0.000000000
Epoch 270: f=   655.439, eps=0.000000000
Epoch 271: f=   655.439, eps=0.000000000
Epoch 272: f=   655.439, eps=0.000000000
Epoch 273: f=   655.439, eps=0.000000000
Epoch 274: f=   655.439, eps=0.000000000
Epoch 275: f=   655.439, eps=0.000000000
Epoch 276: f=   655.439, eps=0.000000000
Epoch 277: f=   655.439, eps=0.000000000
Epoch 278: f=   655.439, eps=0.000000000
Epoch 279: f=   655.439, eps=0.000000000
Epoch 280: f=   655.439, eps=0.000000000
Epoch 281: f=   655.439, eps=0.000000000
Epoch 282: f=   655.439, eps=0.000000000
Epoch 283: f=   655.439, eps=0.000000000
Epoch 284: f=   655.439, eps=0.000000000
Epoch 285: f=   655.439, eps=0.000000000
Epoch 286: f=   655.439, eps=0.000000000
Epoch 287: f=   655.439, eps=0.000000000
Epoch 288: f=   655.439, eps=0.000000000
Epoch 289: f=   655.439, eps=0.000000000
Epoch 290: f=   655.439, eps=0.000000000
Epoch 291: f=   655.439, eps=0.000000000
Epoch 292: f=   655.439, eps=0.000000000
Epoch 293: f=   655.439, eps=0.000000000
Epoch 294: f=   655.439, eps=0.000000000
Epoch 295: f=   655.439, eps=0.000000000
Epoch 296: f=   655.439, eps=0.000000000
Epoch 297: f=   655.439, eps=0.000000000
Epoch 298: f=   655.439, eps=0.000000000
Epoch 299: f=   655.439, eps=0.000000000
Epoch 300: f=   655.439, eps=0.000000000
Epoch 301: f=   655.439, eps=0.000000000
Epoch 302: f=   655.439, eps=0.000000000
Epoch 303: f=   655.439, eps=0.000000000
Epoch 304: f=   655.439, eps=0.000000000
Epoch 305: f=   655.439, eps=0.000000000
Epoch 306: f=   655.439, eps=0.000000000
Epoch 307: f=   655.439, eps=0.000000000
Epoch 308: f=   655.439, eps=0.000000000
Epoch 309: f=   655.439, eps=0.000000000
Epoch 310: f=   655.439, eps=0.000000000
Epoch 311: f=   655.439, eps=0.000000000
Epoch 312: f=   655.439, eps=0.000000000
Epoch 313: f=   655.439, eps=0.000000000
Epoch 314: f=   655.439, eps=0.000000000
Epoch 315: f=   655.439, eps=0.000000000
Epoch 316: f=   655.439, eps=0.000000000
Epoch 317: f=   655.439, eps=0.000000000
Epoch 318: f=   655.439, eps=0.000000000
Epoch 319: f=   655.439, eps=0.000000000
Epoch 320: f=   655.439, eps=0.000000000
Epoch 321: f=   655.439, eps=0.000000000
Epoch 322: f=   655.439, eps=0.000000000
Epoch 323: f=   655.439, eps=0.000000000
Epoch 324: f=   655.439, eps=0.000000000
Epoch 325: f=   655.439, eps=0.000000000
Epoch 326: f=   655.439, eps=0.000000000
Epoch 327: f=   655.439, eps=0.000000000
Epoch 328: f=   655.439, eps=0.000000000
Epoch 329: f=   655.439, eps=0.000000000
Epoch 330: f=   655.439, eps=0.000000000
Epoch 331: f=   655.439, eps=0.000000000
Epoch 332: f=   655.439, eps=0.000000000
Epoch 333: f=   655.439, eps=0.000000000
Epoch 334: f=   655.439, eps=0.000000000
Epoch 335: f=   655.439, eps=0.000000000
Epoch 336: f=   655.439, eps=0.000000000
Epoch 337: f=   655.439, eps=0.000000000
Epoch 338: f=   655.439, eps=0.000000000
Epoch 339: f=   655.439, eps=0.000000000
Epoch 340: f=   655.439, eps=0.000000000
Epoch 341: f=   655.439, eps=0.000000000
Epoch 342: f=   655.439, eps=0.000000000
Epoch 343: f=   655.439, eps=0.000000000
Epoch 344: f=   655.439, eps=0.000000000
Epoch 345: f=   655.439, eps=0.000000000
Epoch 346: f=   655.439, eps=0.000000000
Epoch 347: f=   655.439, eps=0.000000000
Epoch 348: f=   655.439, eps=0.000000000
Epoch 349: f=   655.439, eps=0.000000000
Epoch 350: f=   655.439, eps=0.000000000
Epoch 351: f=   655.439, eps=0.000000000
```

```
Epoch 352: f=   655.439, eps=0.000000000
Epoch 353: f=   655.439, eps=0.000000000
Epoch 354: f=   655.439, eps=0.000000000
Epoch 355: f=   655.439, eps=0.000000000
Epoch 356: f=   655.439, eps=0.000000000
Epoch 357: f=   655.439, eps=0.000000000
Epoch 358: f=   655.439, eps=0.000000000
Epoch 359: f=   655.439, eps=0.000000000
Epoch 360: f=   655.439, eps=0.000000000
Epoch 361: f=   655.439, eps=0.000000000
Epoch 362: f=   655.439, eps=0.000000000
Epoch 363: f=   655.439, eps=0.000000000
Epoch 364: f=   655.439, eps=0.000000000
Epoch 365: f=   655.439, eps=0.000000000
Epoch 366: f=   655.439, eps=0.000000000
Epoch 367: f=   655.439, eps=0.000000000
Epoch 368: f=   655.439, eps=0.000000000
Epoch 369: f=   655.439, eps=0.000000000
Epoch 370: f=   655.439, eps=0.000000000
Epoch 371: f=   655.439, eps=0.000000000
Epoch 372: f=   655.439, eps=0.000000000
Epoch 373: f=   655.439, eps=0.000000000
Epoch 374: f=   655.439, eps=0.000000000
Epoch 375: f=   655.439, eps=0.000000000
Epoch 376: f=   655.439, eps=0.000000000
Epoch 377: f=   655.439, eps=0.000000000
Epoch 378: f=   655.439, eps=0.000000000
Epoch 379: f=   655.439, eps=0.000000000
Epoch 380: f=   655.439, eps=0.000000000
Epoch 381: f=   655.439, eps=0.000000000
Epoch 382: f=   655.439, eps=0.000000000
Epoch 383: f=   655.439, eps=0.000000000
Epoch 384: f=   655.439, eps=0.000000000
Epoch 385: f=   655.439, eps=0.000000000
Epoch 386: f=   655.439, eps=0.000000000
Epoch 387: f=   655.439, eps=0.000000000
Epoch 388: f=   655.439, eps=0.000000000
Epoch 389: f=   655.439, eps=0.000000000
Epoch 390: f=   655.439, eps=0.000000000
Epoch 391: f=   655.439, eps=0.000000000
Epoch 392: f=   655.439, eps=0.000000000
Epoch 393: f=   655.439, eps=0.000000000
Epoch 394: f=   655.439, eps=0.000000000
Epoch 395: f=   655.439, eps=0.000000000
Epoch 396: f=   655.439, eps=0.000000000
Epoch 397: f=   655.439, eps=0.000000000
Epoch 398: f=   655.439, eps=0.000000000
Epoch 399: f=   655.439, eps=0.000000000
Epoch 400: f=   655.439, eps=0.000000000
Epoch 401: f=   655.439, eps=0.000000000
Epoch 402: f=   655.439, eps=0.000000000
Epoch 403: f=   655.439, eps=0.000000000
Epoch 404: f=   655.439, eps=0.000000000
Epoch 405: f=   655.439, eps=0.000000000
Epoch 406: f=   655.439, eps=0.000000000
Epoch 407: f=   655.439, eps=0.000000000
Epoch 408: f=   655.439, eps=0.000000000
Epoch 409: f=   655.439, eps=0.000000000
Epoch 410: f=   655.439, eps=0.000000000
Epoch 411: f=   655.439, eps=0.000000000
Epoch 412: f=   655.439, eps=0.000000000
Epoch 413: f=   655.439, eps=0.000000000
Epoch 414: f=   655.439, eps=0.000000000
Epoch 415: f=   655.439, eps=0.000000000
Epoch 416: f=   655.439, eps=0.000000000
Epoch 417: f=   655.439, eps=0.000000000
Epoch 418: f=   655.439, eps=0.000000000
Epoch 419: f=   655.439, eps=0.000000000
Epoch 420: f=   655.439, eps=0.000000000
Epoch 421: f=   655.439, eps=0.000000000
Epoch 422: f=   655.439, eps=0.000000000
Epoch 423: f=   655.439, eps=0.000000000
Epoch 424: f=   655.439, eps=0.000000000
Epoch 425: f=   655.439, eps=0.000000000
Epoch 426: f=   655.439, eps=0.000000000
Epoch 427: f=   655.439, eps=0.000000000
Epoch 428: f=   655.439, eps=0.000000000
Epoch 429: f=   655.439, eps=0.000000000
Epoch 430: f=   655.439, eps=0.000000000
Epoch 431: f=   655.439, eps=0.000000000
Epoch 432: f=   655.439, eps=0.000000000
Epoch 433: f=   655.439, eps=0.000000000
Epoch 434: f=   655.439, eps=0.000000000
Epoch 435: f=   655.439, eps=0.000000000
Epoch 436: f=   655.439, eps=0.000000000
Epoch 437: f=   655.439, eps=0.000000000
Epoch 438: f=   655.439, eps=0.000000000
Epoch 439: f=   655.439, eps=0.000000000
Epoch 440: f=   655.439, eps=0.000000000
Epoch 441: f=   655.439, eps=0.000000000
Epoch 442: f=   655.439, eps=0.000000000
Epoch 443: f=   655.439, eps=0.000000000
```

```
Epoch 444: f=   655.439, eps=0.000000000
Epoch 445: f=   655.439, eps=0.000000000
Epoch 446: f=   655.439, eps=0.000000000
Epoch 447: f=   655.439, eps=0.000000000
Epoch 448: f=   655.439, eps=0.000000000
Epoch 449: f=   655.439, eps=0.000000000
Epoch 450: f=   655.439, eps=0.000000000
Epoch 451: f=   655.439, eps=0.000000000
Epoch 452: f=   655.439, eps=0.000000000
Epoch 453: f=   655.439, eps=0.000000000
Epoch 454: f=   655.439, eps=0.000000000
Epoch 455: f=   655.439, eps=0.000000000
Epoch 456: f=   655.439, eps=0.000000000
Epoch 457: f=   655.439, eps=0.000000000
Epoch 458: f=   655.439, eps=0.000000000
Epoch 459: f=   655.439, eps=0.000000000
Epoch 460: f=   655.439, eps=0.000000000
Epoch 461: f=   655.439, eps=0.000000000
Epoch 462: f=   655.439, eps=0.000000000
Epoch 463: f=   655.439, eps=0.000000000
Epoch 464: f=   655.439, eps=0.000000000
Epoch 465: f=   655.439, eps=0.000000000
Epoch 466: f=   655.439, eps=0.000000000
Epoch 467: f=   655.439, eps=0.000000000
Epoch 468: f=   655.439, eps=0.000000000
Epoch 469: f=   655.439, eps=0.000000000
Epoch 470: f=   655.439, eps=0.000000000
Epoch 471: f=   655.439, eps=0.000000000
Epoch 472: f=   655.439, eps=0.000000000
Epoch 473: f=   655.439, eps=0.000000000
Epoch 474: f=   655.439, eps=0.000000000
Epoch 475: f=   655.439, eps=0.000000000
Epoch 476: f=   655.439, eps=0.000000000
Epoch 477: f=   655.439, eps=0.000000000
Epoch 478: f=   655.439, eps=0.000000000
Epoch 479: f=   655.439, eps=0.000000000
Epoch 480: f=   655.439, eps=0.000000000
Epoch 481: f=   655.439, eps=0.000000000
Epoch 482: f=   655.439, eps=0.000000000
Epoch 483: f=   655.439, eps=0.000000000
Epoch 484: f=   655.439, eps=0.000000000
Epoch 485: f=   655.439, eps=0.000000000
Epoch 486: f=   655.439, eps=0.000000000
Epoch 487: f=   655.439, eps=0.000000000
Epoch 488: f=   655.439, eps=0.000000000
Epoch 489: f=   655.439, eps=0.000000000
Epoch 490: f=   655.439, eps=0.000000000
Epoch 491: f=   655.439, eps=0.000000000
Epoch 492: f=   655.439, eps=0.000000000
Epoch 493: f=   655.439, eps=0.000000000
Epoch 494: f=   655.439, eps=0.000000000
Epoch 495: f=   655.439, eps=0.000000000
Epoch 496: f=   655.439, eps=0.000000000
Epoch 497: f=   655.439, eps=0.000000000
Epoch 498: f=   655.439, eps=0.000000000
Epoch 499: f=   655.439, eps=0.000000000
Result after 500 epochs: f=655.4392468949942
```

2e Compare GD and SGD

In [31]:

```
# YOUR CODE HERE
nextplot()
plt.plot(range(501),vz_gd,label='Gradient Descent')
plt.plot(range(501),vz_sgd,label='Stochastic Gradient Descent')
plt.legend()
plt.show()
plt.savefig('Figure2.2.png')
```

In [32]:

```
# YOUR CODE HERE
nextplot()
plt.plot(range(501),vz_gd,label='adj Gradient Descent')
plt.plot(range(501),vz_sgd,label='adj Stochastic Gradient Descent')
plt.xlim(0,50)
plt.legend()
plt.show()
plt.savefig('Figure2.3.png')
```

3 Prediction

In [33]:

```
def predict(Xtest, w):
    """Returns vector of predicted confidence values for logistic regression with
weight vector w."""
```

```
    # YOUR CODE HERE
    z = np.dot(Xtest, w)
    prediction = sigma(z)
    return prediction

def classify(Xtest, w):
    """Returns 0/1 vector of predicted class labels for logistic regression with
weight vector w."""
    # YOUR CODE HERE
    return np.where(predict(Xtest,w) > 0.5 , 1, 0)
```

In [34]:
```
# Example: confusion matrix
yhat = predict(Xtestz, wz_gd)
ypred = classify(Xtestz, wz_gd)
print(sklearn.metrics.confusion_matrix(ytest, ypred))  # true x predicted
```

```
[[887  54]
 [ 71 524]]
```

In [35]:
```
# Example: classification report
print(sklearn.metrics.classification_report(ytest, ypred))
```

```
              precision    recall  f1-score   support

           0       0.93      0.94      0.93       941
           1       0.91      0.88      0.89       595

    accuracy                           0.92      1536
   macro avg       0.92      0.91      0.91      1536
weighted avg       0.92      0.92      0.92      1536
```

In [36]:
```
# Example: precision-recall curve (with annotated thresholds)
nextplot()
precision, recall, thresholds = sklearn.metrics.precision_recall_curve(ytest, yhat)
plt.plot(recall, precision)
for x in np.linspace(0, 1, 10, endpoint=False):
    index = int(x * (precision.size - 1))
    plt.text(recall[index], precision[index], "{:3.2f}".format(thresholds[index]))
plt.xlabel("Recall")
plt.ylabel("Precision")
```

Out[36]:
```
Text(0, 0.5, 'Precision')
```

In [37]:
```
# Explore which features are considered important
# YOUR CODE HERE
nextplot()
plt.plot(range(len(wz_gd)),wz_gd,'-o')
plt.xlabel('Index Feature')
plt.ylabel('Weight')
plt.savefig('Figure2.4.png')
```

In [38]:
```
max(wz_gd)
np.argmax(wz_gd)
min(wz_gd)
np.argmin(wz_gd)
```

Out[38]:
```
24
```

4 Maximum Aposteriori Estimation

4a Gradient Descent

In [39]:
```
def l_l2(y, X, w, lambda_):
    """Log-density of posterior of logistic regression with weights w and L2
regularization parameter lambda_"""
    # YOUR CODE HERE
    pos_logistic_regression = l(y,X,w) - (lambda_/2)*(np.dot(w,w))
    #(np.linalg.norm(w,ord=2))
    return pos_logistic_regression
```

In [40]:
```
# this should give:
# [-47066.641667825766, -47312.623810682911]
[l_l2(y, Xz, np.linspace(-5, 5, D), 0), l_l2(y, Xz, np.linspace(-5, 5, D), 1)]
```

Out[40]:
```
[-47066.64166782577, -47312.62381068291]
```

In [41]:
```
def dl_l2(y, X, w, lambda_):
    """Gradient of log-density of posterior of logistic regression with weights w
and L2 regularization parameter lambda_."""
    # YOUR CODE HERE
    gradient = dl(y,X,w) - (lambda_)*w
    return gradient
```

In [42]:
```
# this should give:
# [array([  551.33985842,   143.84116318,   841.83373606,   156.87237578,
#           802.61217579,   795.96202907,   920.69045803,   621.96516752,
#           659.18724769,   470.81259805,   771.32406968,   352.40325626,
#           455.66972482,   234.36600888,   562.45454038,   864.83981264,
#           787.19723703,   649.48042176,   902.6478154 ,   544.00539886,
#          1174.78638035,   120.3598967 ,   839.61141672,   633.30453444,
#          -706.66815087,  -630.2039816 ,  -569.3451386 ,  -527.50996698,
#          -359.53701083,  -476.64334832,  -411.60620464,  -375.11950586,
#          -345.37195689,  -376.22044258,  -407.31761977,  -456.23251936,
#          -596.86960184,  -107.97072355,  -394.82170044,  -229.18125598,
#          -288.46356547,  -362.13402385,  -450.87896465,  -277.03932676,
#          -414.99293368,  -452.28771693,  -167.54649092,  -270.9043748 ,
#          -252.20140951,  -357.72497343,  -259.12468742,   418.35938483,
#           604.54173228,    43.10390907,   152.24258478,   378.16731033,
#           416.12032881]),
#  array([  556.33985842,   148.66259175,   846.4765932 ,   161.33666149,
#           806.89789007,   800.06917193,   924.61902946,   625.71516752,
#           662.75867626,   474.20545519,   774.5383554 ,   355.43897054,
#           458.52686767,   237.04458031,   564.95454038,   867.16124121,
#           789.34009417,   651.44470748,   904.43352968,   545.61254171,
#          1176.21495178,   121.6098967 ,   840.68284529,   634.19739158,
#          -705.95386516,  -629.66826731,  -568.98799574,  -527.33139555,
#          -359.53701083,  -476.82191975,  -411.9633475 ,  -375.65522015,
#          -346.08624261,  -377.11329972,  -408.38904835,  -457.48251936,
#          -598.29817327,  -109.57786641,  -396.60741472,  -231.14554169,
#          -290.60642261,  -364.45545242,  -453.37896465,  -279.71789819,
#          -417.85007654,  -455.32343122,  -170.76077664,  -274.29723194,
#          -255.77283808,  -361.47497343,  -263.05325885,   414.25224198,
#           600.25601799,    38.63962335,   147.59972763,   373.34588176,
#           411.12032881])]
[dl_l2(y, Xz, np.linspace(-5, 5, D), 0), dl_l2(y, Xz, np.linspace(-5, 5, D), 1)]
```

Out[42]:
```
[array([ 551.33985842,  143.84116318,  841.83373606,  156.87237578,
         802.61217579,  795.96202907,  920.69045803,  621.96516752,
         659.18724769,  470.81259805,  771.32406968,  352.40325626,
         455.66972482,  234.36600888,  562.45454038,  864.83981264,
         787.19723703,  649.48042176,  902.6478154 ,  544.00539886,
        1174.78638035,  120.3598967 ,  839.61141672,  633.30453444,
        -706.66815087, -630.2039816 , -569.3451386 , -527.50996698,
        -359.53701083, -476.64334832, -411.60620464, -375.11950586,
        -345.37195689, -376.22044258, -407.31761977, -456.23251936,
        -596.86960184, -107.97072355, -394.82170044, -229.18125598,
        -288.46356547, -362.13402385, -450.87896465, -277.03932676,
        -414.99293368, -452.28771693, -167.54649092, -270.9043748 ,
        -252.20140951, -357.72497343, -259.12468742,  418.35938483,
         604.54173228,   43.10390907,  152.24258478,  378.16731033,
         416.12032881]),
 array([ 556.33985842,  148.66259175,  846.4765932 ,  161.33666149,
         806.89789007,  800.06917193,  924.61902946,  625.71516752,
         662.75867626,  474.20545519,  774.5383554 ,  355.43897054,
         458.52686767,  237.04458031,  564.95454038,  867.16124121,
         789.34009417,  651.44470748,  904.43352968,  545.61254171,
        1176.21495178,  121.6098967 ,  840.68284529,  634.19739158,
        -705.95386516, -629.66826731, -568.98799574, -527.33139555,
        -359.53701083, -476.82191975, -411.9633475 , -375.65522015,
        -346.08624261, -377.11329972, -408.38904835, -457.48251936,
        -598.29817327, -109.57786641, -396.60741472, -231.14554169,
        -290.60642261, -364.45545242, -453.37896465, -279.71789819,
        -417.85007654, -455.32343122, -170.76077664, -274.29723194,
        -255.77283808, -361.47497343, -263.05325885,  414.25224198,
         600.25601799,   38.63962335,  147.59972763,  373.34588176,
         411.12032881])]
```

In [43]:
```
# now define the (f,update) tuple for optimize for logistic regression, L2
# regularization, and gradient descent
def gd_l2(y, X, lambda_):
    # YOUR CODE HERE
    def objective(w):
        p_i0 = logsigma(-np.dot(X,w))
        p_i1 = logsigma(np.dot(X,w))
        objective = -np.sum(y*p_i1 + (1-y)*p_i0)
        l2 = (lambda_/2)*(np.dot(w,w))
        return objective+l2
    def update(w,eps) :
        gradient = dl_l2(y,X,w,lambda_)
        w += eps * gradient
        return w
    return (objective,update)
```

In [44]:
```
# let's run!
```

```
lambda_ = 100
wz_gd_l2, vz_gd_l2, ez_gd_l2 = optimize(gd_l2(y, Xz, lambda_), w0, nepochs=500)
```

```
Epoch   0: f=  5484.455, eps=0.010000000
Epoch   1: f=  2137.652, eps=0.010500000
Epoch   2: f= 30782.824, eps=0.005250000
Epoch   3: f=  6484.999, eps=0.005512500
Epoch   4: f=  1504.659, eps=0.005788125
Epoch   5: f=  1141.295, eps=0.006077531
Epoch   6: f=  1771.465, eps=0.003038766
Epoch   7: f=  1585.487, eps=0.003190704
Epoch   8: f=  1075.240, eps=0.003350239
Epoch   9: f=  1073.052, eps=0.003517751
Epoch  10: f=  1116.047, eps=0.001758876
Epoch  11: f=  1017.587, eps=0.001846819
Epoch  12: f=   990.329, eps=0.001939160
Epoch  13: f=   988.737, eps=0.002036118
Epoch  14: f=   988.563, eps=0.002137924
Epoch  15: f=   988.528, eps=0.002244820
Epoch  16: f=   988.518, eps=0.002357061
Epoch  17: f=   988.514, eps=0.002474914
Epoch  18: f=   988.513, eps=0.002598660
Epoch  19: f=   988.513, eps=0.002728593
Epoch  20: f=   988.513, eps=0.002865023
Epoch  21: f=   988.513, eps=0.001432511
Epoch  22: f=   988.512, eps=0.001504137
Epoch  23: f=   988.512, eps=0.001579344
Epoch  24: f=   988.512, eps=0.001658311
Epoch  25: f=   988.512, eps=0.001741227
Epoch  26: f=   988.512, eps=0.001828288
Epoch  27: f=   988.512, eps=0.001919702
Epoch  28: f=   988.512, eps=0.002015687
Epoch  29: f=   988.512, eps=0.002116472
Epoch  30: f=   988.512, eps=0.002222295
Epoch  31: f=   988.512, eps=0.002333410
Epoch  32: f=   988.512, eps=0.002450081
Epoch  33: f=   988.512, eps=0.002572585
Epoch  34: f=   988.512, eps=0.002701214
Epoch  35: f=   988.512, eps=0.002836275
Epoch  36: f=   988.512, eps=0.002978088
Epoch  37: f=   988.512, eps=0.003126993
Epoch  38: f=   988.512, eps=0.003283342
Epoch  39: f=   988.512, eps=0.003447510
Epoch  40: f=   988.512, eps=0.003619885
Epoch  41: f=   988.512, eps=0.003800879
Epoch  42: f=   988.512, eps=0.003990923
Epoch  43: f=   988.512, eps=0.004190469
Epoch  44: f=   988.512, eps=0.002095235
Epoch  45: f=   988.512, eps=0.002199996
Epoch  46: f=   988.512, eps=0.002309996
Epoch  47: f=   988.512, eps=0.002425496
Epoch  48: f=   988.512, eps=0.002546771
Epoch  49: f=   988.512, eps=0.001273385
Epoch  50: f=   988.512, eps=0.001337055
Epoch  51: f=   988.512, eps=0.001403907
Epoch  52: f=   988.512, eps=0.001474103
Epoch  53: f=   988.512, eps=0.001547808
Epoch  54: f=   988.512, eps=0.000773904
Epoch  55: f=   988.512, eps=0.000812599
Epoch  56: f=   988.512, eps=0.000406300
Epoch  57: f=   988.512, eps=0.000426615
Epoch  58: f=   988.512, eps=0.000213307
Epoch  59: f=   988.512, eps=0.000223973
Epoch  60: f=   988.512, eps=0.000235171
Epoch  61: f=   988.512, eps=0.000246930
Epoch  62: f=   988.512, eps=0.000123465
Epoch  63: f=   988.512, eps=0.000129638
Epoch  64: f=   988.512, eps=0.000136120
Epoch  65: f=   988.512, eps=0.000068060
Epoch  66: f=   988.512, eps=0.000071463
Epoch  67: f=   988.512, eps=0.000075036
Epoch  68: f=   988.512, eps=0.000078788
Epoch  69: f=   988.512, eps=0.000039394
Epoch  70: f=   988.512, eps=0.000041364
Epoch  71: f=   988.512, eps=0.000043432
Epoch  72: f=   988.512, eps=0.000045603
Epoch  73: f=   988.512, eps=0.000022802
Epoch  74: f=   988.512, eps=0.000023942
Epoch  75: f=   988.512, eps=0.000011971
Epoch  76: f=   988.512, eps=0.000012569
Epoch  77: f=   988.512, eps=0.000013198
Epoch  78: f=   988.512, eps=0.000013858
Epoch  79: f=   988.512, eps=0.000006929
Epoch  80: f=   988.512, eps=0.000003464
Epoch  81: f=   988.512, eps=0.000003638
Epoch  82: f=   988.512, eps=0.000001819
Epoch  83: f=   988.512, eps=0.000001910
Epoch  84: f=   988.512, eps=0.000000955
Epoch  85: f=   988.512, eps=0.000001003
Epoch  86: f=   988.512, eps=0.000000501
Epoch  87: f=   988.512, eps=0.000000526
```

```
Epoch  88: f=   988.512, eps=0.000000553
Epoch  89: f=   988.512, eps=0.000000276
Epoch  90: f=   988.512, eps=0.000000290
Epoch  91: f=   988.512, eps=0.000000305
Epoch  92: f=   988.512, eps=0.000000320
Epoch  93: f=   988.512, eps=0.000000160
Epoch  94: f=   988.512, eps=0.000000168
Epoch  95: f=   988.512, eps=0.000000176
Epoch  96: f=   988.512, eps=0.000000185
Epoch  97: f=   988.512, eps=0.000000093
Epoch  98: f=   988.512, eps=0.000000097
Epoch  99: f=   988.512, eps=0.000000102
Epoch 100: f=   988.512, eps=0.000000051
Epoch 101: f=   988.512, eps=0.000000054
Epoch 102: f=   988.512, eps=0.000000027
Epoch 103: f=   988.512, eps=0.000000028
Epoch 104: f=   988.512, eps=0.000000030
Epoch 105: f=   988.512, eps=0.000000031
Epoch 106: f=   988.512, eps=0.000000033
Epoch 107: f=   988.512, eps=0.000000034
Epoch 108: f=   988.512, eps=0.000000036
Epoch 109: f=   988.512, eps=0.000000038
Epoch 110: f=   988.512, eps=0.000000040
Epoch 111: f=   988.512, eps=0.000000042
Epoch 112: f=   988.512, eps=0.000000044
Epoch 113: f=   988.512, eps=0.000000022
Epoch 114: f=   988.512, eps=0.000000023
Epoch 115: f=   988.512, eps=0.000000024
Epoch 116: f=   988.512, eps=0.000000025
Epoch 117: f=   988.512, eps=0.000000013
Epoch 118: f=   988.512, eps=0.000000013
Epoch 119: f=   988.512, eps=0.000000007
Epoch 120: f=   988.512, eps=0.000000007
Epoch 121: f=   988.512, eps=0.000000007
Epoch 122: f=   988.512, eps=0.000000008
Epoch 123: f=   988.512, eps=0.000000008
Epoch 124: f=   988.512, eps=0.000000008
Epoch 125: f=   988.512, eps=0.000000004
Epoch 126: f=   988.512, eps=0.000000004
Epoch 127: f=   988.512, eps=0.000000002
Epoch 128: f=   988.512, eps=0.000000002
Epoch 129: f=   988.512, eps=0.000000002
Epoch 130: f=   988.512, eps=0.000000003
Epoch 131: f=   988.512, eps=0.000000001
Epoch 132: f=   988.512, eps=0.000000001
Epoch 133: f=   988.512, eps=0.000000001
Epoch 134: f=   988.512, eps=0.000000001
Epoch 135: f=   988.512, eps=0.000000002
Epoch 136: f=   988.512, eps=0.000000002
Epoch 137: f=   988.512, eps=0.000000002
Epoch 138: f=   988.512, eps=0.000000002
Epoch 139: f=   988.512, eps=0.000000002
Epoch 140: f=   988.512, eps=0.000000002
Epoch 141: f=   988.512, eps=0.000000002
Epoch 142: f=   988.512, eps=0.000000001
Epoch 143: f=   988.512, eps=0.000000001
Epoch 144: f=   988.512, eps=0.000000001
Epoch 145: f=   988.512, eps=0.000000001
Epoch 146: f=   988.512, eps=0.000000001
Epoch 147: f=   988.512, eps=0.000000001
Epoch 148: f=   988.512, eps=0.000000001
Epoch 149: f=   988.512, eps=0.000000001
Epoch 150: f=   988.512, eps=0.000000001
Epoch 151: f=   988.512, eps=0.000000001
Epoch 152: f=   988.512, eps=0.000000001
Epoch 153: f=   988.512, eps=0.000000000
Epoch 154: f=   988.512, eps=0.000000000
Epoch 155: f=   988.512, eps=0.000000000
Epoch 156: f=   988.512, eps=0.000000000
Epoch 157: f=   988.512, eps=0.000000000
Epoch 158: f=   988.512, eps=0.000000001
Epoch 159: f=   988.512, eps=0.000000000
Epoch 160: f=   988.512, eps=0.000000000
Epoch 161: f=   988.512, eps=0.000000000
Epoch 162: f=   988.512, eps=0.000000000
Epoch 163: f=   988.512, eps=0.000000000
Epoch 164: f=   988.512, eps=0.000000000
Epoch 165: f=   988.512, eps=0.000000000
Epoch 166: f=   988.512, eps=0.000000000
Epoch 167: f=   988.512, eps=0.000000000
Epoch 168: f=   988.512, eps=0.000000000
Epoch 169: f=   988.512, eps=0.000000000
Epoch 170: f=   988.512, eps=0.000000000
Epoch 171: f=   988.512, eps=0.000000000
Epoch 172: f=   988.512, eps=0.000000000
Epoch 173: f=   988.512, eps=0.000000001
Epoch 174: f=   988.512, eps=0.000000001
Epoch 175: f=   988.512, eps=0.000000001
Epoch 176: f=   988.512, eps=0.000000001
Epoch 177: f=   988.512, eps=0.000000001
Epoch 178: f=   988.512, eps=0.000000001
Epoch 179: f=   988.512, eps=0.000000001
```

```
Epoch 180: f=   988.512, eps=0.000000001
Epoch 181: f=   988.512, eps=0.000000001
Epoch 182: f=   988.512, eps=0.000000001
Epoch 183: f=   988.512, eps=0.000000001
Epoch 184: f=   988.512, eps=0.000000001
Epoch 185: f=   988.512, eps=0.000000001
Epoch 186: f=   988.512, eps=0.000000001
Epoch 187: f=   988.512, eps=0.000000001
Epoch 188: f=   988.512, eps=0.000000001
Epoch 189: f=   988.512, eps=0.000000001
Epoch 190: f=   988.512, eps=0.000000001
Epoch 191: f=   988.512, eps=0.000000001
Epoch 192: f=   988.512, eps=0.000000001
Epoch 193: f=   988.512, eps=0.000000001
Epoch 194: f=   988.512, eps=0.000000001
Epoch 195: f=   988.512, eps=0.000000001
Epoch 196: f=   988.512, eps=0.000000001
Epoch 197: f=   988.512, eps=0.000000001
Epoch 198: f=   988.512, eps=0.000000000
Epoch 199: f=   988.512, eps=0.000000000
Epoch 200: f=   988.512, eps=0.000000000
Epoch 201: f=   988.512, eps=0.000000000
Epoch 202: f=   988.512, eps=0.000000000
Epoch 203: f=   988.512, eps=0.000000001
Epoch 204: f=   988.512, eps=0.000000001
Epoch 205: f=   988.512, eps=0.000000001
Epoch 206: f=   988.512, eps=0.000000001
Epoch 207: f=   988.512, eps=0.000000001
Epoch 208: f=   988.512, eps=0.000000001
Epoch 209: f=   988.512, eps=0.000000001
Epoch 210: f=   988.512, eps=0.000000001
Epoch 211: f=   988.512, eps=0.000000001
Epoch 212: f=   988.512, eps=0.000000001
Epoch 213: f=   988.512, eps=0.000000001
Epoch 214: f=   988.512, eps=0.000000001
Epoch 215: f=   988.512, eps=0.000000001
Epoch 216: f=   988.512, eps=0.000000001
Epoch 217: f=   988.512, eps=0.000000001
Epoch 218: f=   988.512, eps=0.000000001
Epoch 219: f=   988.512, eps=0.000000001
Epoch 220: f=   988.512, eps=0.000000001
Epoch 221: f=   988.512, eps=0.000000001
Epoch 222: f=   988.512, eps=0.000000001
Epoch 223: f=   988.512, eps=0.000000001
Epoch 224: f=   988.512, eps=0.000000001
Epoch 225: f=   988.512, eps=0.000000001
Epoch 226: f=   988.512, eps=0.000000000
Epoch 227: f=   988.512, eps=0.000000000
Epoch 228: f=   988.512, eps=0.000000000
Epoch 229: f=   988.512, eps=0.000000000
Epoch 230: f=   988.512, eps=0.000000000
Epoch 231: f=   988.512, eps=0.000000000
Epoch 232: f=   988.512, eps=0.000000000
Epoch 233: f=   988.512, eps=0.000000000
Epoch 234: f=   988.512, eps=0.000000000
Epoch 235: f=   988.512, eps=0.000000000
Epoch 236: f=   988.512, eps=0.000000000
Epoch 237: f=   988.512, eps=0.000000000
Epoch 238: f=   988.512, eps=0.000000000
Epoch 239: f=   988.512, eps=0.000000000
Epoch 240: f=   988.512, eps=0.000000000
Epoch 241: f=   988.512, eps=0.000000000
Epoch 242: f=   988.512, eps=0.000000000
Epoch 243: f=   988.512, eps=0.000000000
Epoch 244: f=   988.512, eps=0.000000000
Epoch 245: f=   988.512, eps=0.000000000
Epoch 246: f=   988.512, eps=0.000000000
Epoch 247: f=   988.512, eps=0.000000000
Epoch 248: f=   988.512, eps=0.000000000
Epoch 249: f=   988.512, eps=0.000000000
Epoch 250: f=   988.512, eps=0.000000000
Epoch 251: f=   988.512, eps=0.000000000
Epoch 252: f=   988.512, eps=0.000000000
Epoch 253: f=   988.512, eps=0.000000000
Epoch 254: f=   988.512, eps=0.000000000
Epoch 255: f=   988.512, eps=0.000000000
Epoch 256: f=   988.512, eps=0.000000000
Epoch 257: f=   988.512, eps=0.000000000
Epoch 258: f=   988.512, eps=0.000000000
Epoch 259: f=   988.512, eps=0.000000000
Epoch 260: f=   988.512, eps=0.000000000
Epoch 261: f=   988.512, eps=0.000000000
Epoch 262: f=   988.512, eps=0.000000000
Epoch 263: f=   988.512, eps=0.000000000
Epoch 264: f=   988.512, eps=0.000000000
Epoch 265: f=   988.512, eps=0.000000000
Epoch 266: f=   988.512, eps=0.000000000
Epoch 267: f=   988.512, eps=0.000000000
Epoch 268: f=   988.512, eps=0.000000000
Epoch 269: f=   988.512, eps=0.000000000
Epoch 270: f=   988.512, eps=0.000000000
Epoch 271: f=   988.512, eps=0.000000000
```

```
Epoch 272: f=   988.512, eps=0.000000000
Epoch 273: f=   988.512, eps=0.000000000
Epoch 274: f=   988.512, eps=0.000000000
Epoch 275: f=   988.512, eps=0.000000000
Epoch 276: f=   988.512, eps=0.000000000
Epoch 277: f=   988.512, eps=0.000000000
Epoch 278: f=   988.512, eps=0.000000000
Epoch 279: f=   988.512, eps=0.000000000
Epoch 280: f=   988.512, eps=0.000000000
Epoch 281: f=   988.512, eps=0.000000000
Epoch 282: f=   988.512, eps=0.000000000
Epoch 283: f=   988.512, eps=0.000000000
Epoch 284: f=   988.512, eps=0.000000000
Epoch 285: f=   988.512, eps=0.000000000
Epoch 286: f=   988.512, eps=0.000000000
Epoch 287: f=   988.512, eps=0.000000000
Epoch 288: f=   988.512, eps=0.000000000
Epoch 289: f=   988.512, eps=0.000000000
Epoch 290: f=   988.512, eps=0.000000000
Epoch 291: f=   988.512, eps=0.000000000
Epoch 292: f=   988.512, eps=0.000000000
Epoch 293: f=   988.512, eps=0.000000000
Epoch 294: f=   988.512, eps=0.000000000
Epoch 295: f=   988.512, eps=0.000000001
Epoch 296: f=   988.512, eps=0.000000001
Epoch 297: f=   988.512, eps=0.000000001
Epoch 298: f=   988.512, eps=0.000000001
Epoch 299: f=   988.512, eps=0.000000001
Epoch 300: f=   988.512, eps=0.000000001
Epoch 301: f=   988.512, eps=0.000000001
Epoch 302: f=   988.512, eps=0.000000000
Epoch 303: f=   988.512, eps=0.000000000
Epoch 304: f=   988.512, eps=0.000000000
Epoch 305: f=   988.512, eps=0.000000000
Epoch 306: f=   988.512, eps=0.000000000
Epoch 307: f=   988.512, eps=0.000000000
Epoch 308: f=   988.512, eps=0.000000000
Epoch 309: f=   988.512, eps=0.000000000
Epoch 310: f=   988.512, eps=0.000000001
Epoch 311: f=   988.512, eps=0.000000001
Epoch 312: f=   988.512, eps=0.000000001
Epoch 313: f=   988.512, eps=0.000000001
Epoch 314: f=   988.512, eps=0.000000001
Epoch 315: f=   988.512, eps=0.000000001
Epoch 316: f=   988.512, eps=0.000000001
Epoch 317: f=   988.512, eps=0.000000000
Epoch 318: f=   988.512, eps=0.000000000
Epoch 319: f=   988.512, eps=0.000000000
Epoch 320: f=   988.512, eps=0.000000000
Epoch 321: f=   988.512, eps=0.000000000
Epoch 322: f=   988.512, eps=0.000000000
Epoch 323: f=   988.512, eps=0.000000000
Epoch 324: f=   988.512, eps=0.000000000
Epoch 325: f=   988.512, eps=0.000000001
Epoch 326: f=   988.512, eps=0.000000001
Epoch 327: f=   988.512, eps=0.000000001
Epoch 328: f=   988.512, eps=0.000000001
Epoch 329: f=   988.512, eps=0.000000001
Epoch 330: f=   988.512, eps=0.000000001
Epoch 331: f=   988.512, eps=0.000000001
Epoch 332: f=   988.512, eps=0.000000000
Epoch 333: f=   988.512, eps=0.000000000
Epoch 334: f=   988.512, eps=0.000000000
Epoch 335: f=   988.512, eps=0.000000000
Epoch 336: f=   988.512, eps=0.000000000
Epoch 337: f=   988.512, eps=0.000000000
Epoch 338: f=   988.512, eps=0.000000000
Epoch 339: f=   988.512, eps=0.000000000
Epoch 340: f=   988.512, eps=0.000000001
Epoch 341: f=   988.512, eps=0.000000000
Epoch 342: f=   988.512, eps=0.000000000
Epoch 343: f=   988.512, eps=0.000000000
Epoch 344: f=   988.512, eps=0.000000000
Epoch 345: f=   988.512, eps=0.000000000
Epoch 346: f=   988.512, eps=0.000000000
Epoch 347: f=   988.512, eps=0.000000000
Epoch 348: f=   988.512, eps=0.000000000
Epoch 349: f=   988.512, eps=0.000000000
Epoch 350: f=   988.512, eps=0.000000000
Epoch 351: f=   988.512, eps=0.000000000
Epoch 352: f=   988.512, eps=0.000000000
Epoch 353: f=   988.512, eps=0.000000000
Epoch 354: f=   988.512, eps=0.000000000
Epoch 355: f=   988.512, eps=0.000000000
Epoch 356: f=   988.512, eps=0.000000000
Epoch 357: f=   988.512, eps=0.000000000
Epoch 358: f=   988.512, eps=0.000000000
Epoch 359: f=   988.512, eps=0.000000000
Epoch 360: f=   988.512, eps=0.000000000
Epoch 361: f=   988.512, eps=0.000000000
Epoch 362: f=   988.512, eps=0.000000000
Epoch 363: f=   988.512, eps=0.000000000
```

```
Epoch 364: f=   988.512, eps=0.000000000
Epoch 365: f=   988.512, eps=0.000000000
Epoch 366: f=   988.512, eps=0.000000000
Epoch 367: f=   988.512, eps=0.000000000
Epoch 368: f=   988.512, eps=0.000000000
Epoch 369: f=   988.512, eps=0.000000000
Epoch 370: f=   988.512, eps=0.000000000
Epoch 371: f=   988.512, eps=0.000000000
Epoch 372: f=   988.512, eps=0.000000000
Epoch 373: f=   988.512, eps=0.000000000
Epoch 374: f=   988.512, eps=0.000000000
Epoch 375: f=   988.512, eps=0.000000000
Epoch 376: f=   988.512, eps=0.000000000
Epoch 377: f=   988.512, eps=0.000000000
Epoch 378: f=   988.512, eps=0.000000000
Epoch 379: f=   988.512, eps=0.000000000
Epoch 380: f=   988.512, eps=0.000000000
Epoch 381: f=   988.512, eps=0.000000000
Epoch 382: f=   988.512, eps=0.000000000
Epoch 383: f=   988.512, eps=0.000000000
Epoch 384: f=   988.512, eps=0.000000000
Epoch 385: f=   988.512, eps=0.000000000
Epoch 386: f=   988.512, eps=0.000000000
Epoch 387: f=   988.512, eps=0.000000000
Epoch 388: f=   988.512, eps=0.000000000
Epoch 389: f=   988.512, eps=0.000000000
Epoch 390: f=   988.512, eps=0.000000000
Epoch 391: f=   988.512, eps=0.000000000
Epoch 392: f=   988.512, eps=0.000000000
Epoch 393: f=   988.512, eps=0.000000000
Epoch 394: f=   988.512, eps=0.000000000
Epoch 395: f=   988.512, eps=0.000000000
Epoch 396: f=   988.512, eps=0.000000000
Epoch 397: f=   988.512, eps=0.000000000
Epoch 398: f=   988.512, eps=0.000000000
Epoch 399: f=   988.512, eps=0.000000000
Epoch 400: f=   988.512, eps=0.000000000
Epoch 401: f=   988.512, eps=0.000000000
Epoch 402: f=   988.512, eps=0.000000000
Epoch 403: f=   988.512, eps=0.000000000
Epoch 404: f=   988.512, eps=0.000000000
Epoch 405: f=   988.512, eps=0.000000000
Epoch 406: f=   988.512, eps=0.000000000
Epoch 407: f=   988.512, eps=0.000000000
Epoch 408: f=   988.512, eps=0.000000000
Epoch 409: f=   988.512, eps=0.000000000
Epoch 410: f=   988.512, eps=0.000000000
Epoch 411: f=   988.512, eps=0.000000000
Epoch 412: f=   988.512, eps=0.000000000
Epoch 413: f=   988.512, eps=0.000000000
Epoch 414: f=   988.512, eps=0.000000000
Epoch 415: f=   988.512, eps=0.000000000
Epoch 416: f=   988.512, eps=0.000000000
Epoch 417: f=   988.512, eps=0.000000000
Epoch 418: f=   988.512, eps=0.000000000
Epoch 419: f=   988.512, eps=0.000000000
Epoch 420: f=   988.512, eps=0.000000000
Epoch 421: f=   988.512, eps=0.000000000
Epoch 422: f=   988.512, eps=0.000000000
Epoch 423: f=   988.512, eps=0.000000000
Epoch 424: f=   988.512, eps=0.000000000
Epoch 425: f=   988.512, eps=0.000000000
Epoch 426: f=   988.512, eps=0.000000000
Epoch 427: f=   988.512, eps=0.000000000
Epoch 428: f=   988.512, eps=0.000000000
Epoch 429: f=   988.512, eps=0.000000000
Epoch 430: f=   988.512, eps=0.000000000
Epoch 431: f=   988.512, eps=0.000000000
Epoch 432: f=   988.512, eps=0.000000000
Epoch 433: f=   988.512, eps=0.000000000
Epoch 434: f=   988.512, eps=0.000000000
Epoch 435: f=   988.512, eps=0.000000000
Epoch 436: f=   988.512, eps=0.000000000
Epoch 437: f=   988.512, eps=0.000000000
Epoch 438: f=   988.512, eps=0.000000000
Epoch 439: f=   988.512, eps=0.000000000
Epoch 440: f=   988.512, eps=0.000000000
Epoch 441: f=   988.512, eps=0.000000000
Epoch 442: f=   988.512, eps=0.000000000
Epoch 443: f=   988.512, eps=0.000000000
Epoch 444: f=   988.512, eps=0.000000000
Epoch 445: f=   988.512, eps=0.000000000
Epoch 446: f=   988.512, eps=0.000000000
Epoch 447: f=   988.512, eps=0.000000000
Epoch 448: f=   988.512, eps=0.000000000
Epoch 449: f=   988.512, eps=0.000000000
Epoch 450: f=   988.512, eps=0.000000000
Epoch 451: f=   988.512, eps=0.000000000
Epoch 452: f=   988.512, eps=0.000000000
Epoch 453: f=   988.512, eps=0.000000000
Epoch 454: f=   988.512, eps=0.000000000
Epoch 455: f=   988.512, eps=0.000000000
```

```
Epoch 456: f=   988.512, eps=0.000000000
Epoch 457: f=   988.512, eps=0.000000000
Epoch 458: f=   988.512, eps=0.000000000
Epoch 459: f=   988.512, eps=0.000000000
Epoch 460: f=   988.512, eps=0.000000000
Epoch 461: f=   988.512, eps=0.000000000
Epoch 462: f=   988.512, eps=0.000000000
Epoch 463: f=   988.512, eps=0.000000000
Epoch 464: f=   988.512, eps=0.000000000
Epoch 465: f=   988.512, eps=0.000000000
Epoch 466: f=   988.512, eps=0.000000000
Epoch 467: f=   988.512, eps=0.000000000
Epoch 468: f=   988.512, eps=0.000000000
Epoch 469: f=   988.512, eps=0.000000000
Epoch 470: f=   988.512, eps=0.000000000
Epoch 471: f=   988.512, eps=0.000000000
Epoch 472: f=   988.512, eps=0.000000000
Epoch 473: f=   988.512, eps=0.000000000
Epoch 474: f=   988.512, eps=0.000000000
Epoch 475: f=   988.512, eps=0.000000000
Epoch 476: f=   988.512, eps=0.000000000
Epoch 477: f=   988.512, eps=0.000000000
Epoch 478: f=   988.512, eps=0.000000000
Epoch 479: f=   988.512, eps=0.000000000
Epoch 480: f=   988.512, eps=0.000000000
Epoch 481: f=   988.512, eps=0.000000000
Epoch 482: f=   988.512, eps=0.000000000
Epoch 483: f=   988.512, eps=0.000000000
Epoch 484: f=   988.512, eps=0.000000000
Epoch 485: f=   988.512, eps=0.000000000
Epoch 486: f=   988.512, eps=0.000000000
Epoch 487: f=   988.512, eps=0.000000000
Epoch 488: f=   988.512, eps=0.000000000
Epoch 489: f=   988.512, eps=0.000000000
Epoch 490: f=   988.512, eps=0.000000000
Epoch 491: f=   988.512, eps=0.000000000
Epoch 492: f=   988.512, eps=0.000000000
Epoch 493: f=   988.512, eps=0.000000000
Epoch 494: f=   988.512, eps=0.000000000
Epoch 495: f=   988.512, eps=0.000000000
Epoch 496: f=   988.512, eps=0.000000000
Epoch 497: f=   988.512, eps=0.000000000
Epoch 498: f=   988.512, eps=0.000000000
Epoch 499: f=   988.512, eps=0.000000000
Result after 500 epochs: f=988.511839602703
```

4b Effect of Prior

In [45]:

```
# YOUR CODE HERE
```

4c Composition of Weight Vector

In [46]:

```
# YOUR CODE HERE
```

5 Exploration (optional)

5 Exploration: PyTorch

In [47]:

```
# if you want to experiment, here is an implementation of logistic
# regression in PyTorch
import math
import torch
import torch.nn as nn
import torch.utils.data
import torch.nn.functional as F

# prepare the data
Xztorch = torch.FloatTensor(Xz)
ytorch = torch.LongTensor(y)
train = torch.utils.data.TensorDataset(Xztorch, ytorch)


# manual implementation of logistic regression (without bias)
class LogisticRegression(nn.Module):
    def __init__(self, D, C):
        super(LogisticRegression, self).__init__()
        self.weights = torch.nn.Parameter(
            torch.randn(D, C) / math.sqrt(D)
        )  # xavier initialization
        self.register_parameter("W", self.weights)

    def forward(self, x):
        out = torch.matmul(x, self.weights)
        out = F.log_softmax(out)
        return out
```

```
# define the objective and update function. here we ignore the learning rates
# and parameters given to us by optimize (they are stored in the PyTorch model
# and optimizer, resp., instead)
def opt_pytorch():
    model = LogisticRegression(D, 2)
    criterion = nn.NLLLoss(reduction="sum")
    # change the next line to try different optimizers
    # optimizer = torch.optim.SGD(model.parameters(), lr=learning_rate)
    optimizer = torch.optim.Adam(model.parameters(), lr=learning_rate)

    def objective(_):
        outputs = model(Xztorch)
        return criterion(outputs, ytorch)

    def update(_1, _2):
        for i, (examples, labels) in enumerate(train_loader):
            outputs = model(examples)
            loss = criterion(outputs, labels)
            optimizer.zero_grad()
            loss.backward()
            optimizer.step()

        W = model.state_dict()["W"]
        w = W[:, 1] - W[:, 0]
        return w

    return (objective, update)
```

In [48]:

```
# run the optimizer
learning_rate = 0.01
batch_size = 100  # number of data points to sample for gradient estimate
shuffle = True  # sample with replacement (false) or without replacement (true)

train_loader = torch.utils.data.DataLoader(train, batch_size, shuffle=True)
wz_t, vz_t, _ = optimize(opt_pytorch(), None, nepochs=100, eps0=None, verbose=True)
```

```
/var/folders/j5/tqm3_jyd1mz9mmb920s62hlm0000gn/T/ipykernel_31523/2194961090.py:26: UserWarning: Implicit dimension ch
oice for log_softmax has been deprecated. Change the call to include dim=X as an argument.
  out = F.log_softmax(out)
Epoch   0: f=  2815.387, eps=       nan
Epoch   1: f=   907.160, eps=       nan
Epoch   2: f=   791.752, eps=       nan
Epoch   3: f=   749.509, eps=       nan
Epoch   4: f=   726.905, eps=       nan
Epoch   5: f=   712.422, eps=       nan
Epoch   6: f=   700.858, eps=       nan
Epoch   7: f=   693.756, eps=       nan
Epoch   8: f=   689.438, eps=       nan
Epoch   9: f=   683.860, eps=       nan
Epoch  10: f=   680.829, eps=       nan
Epoch  11: f=   678.680, eps=       nan
Epoch  12: f=   676.080, eps=       nan
Epoch  13: f=   673.790, eps=       nan
Epoch  14: f=   672.939, eps=       nan
Epoch  15: f=   671.321, eps=       nan
Epoch  16: f=   670.018, eps=       nan
Epoch  17: f=   668.922, eps=       nan
Epoch  18: f=   668.005, eps=       nan
Epoch  19: f=   667.138, eps=       nan
Epoch  20: f=   666.417, eps=       nan
Epoch  21: f=   665.473, eps=       nan
Epoch  22: f=   664.786, eps=       nan
Epoch  23: f=   664.258, eps=       nan
Epoch  24: f=   664.024, eps=       nan
Epoch  25: f=   663.821, eps=       nan
Epoch  26: f=   662.374, eps=       nan
Epoch  27: f=   661.638, eps=       nan
Epoch  28: f=   661.149, eps=       nan
Epoch  29: f=   660.570, eps=       nan
Epoch  30: f=   659.699, eps=       nan
Epoch  31: f=   659.075, eps=       nan
Epoch  32: f=   658.703, eps=       nan
Epoch  33: f=   659.024, eps=       nan
Epoch  34: f=   658.135, eps=       nan
Epoch  35: f=   657.954, eps=       nan
Epoch  36: f=   657.135, eps=       nan
Epoch  37: f=   656.981, eps=       nan
Epoch  38: f=   661.007, eps=       nan
Epoch  39: f=   656.330, eps=       nan
Epoch  40: f=   655.512, eps=       nan
Epoch  41: f=   656.052, eps=       nan
Epoch  42: f=   656.766, eps=       nan
Epoch  43: f=   654.745, eps=       nan
Epoch  44: f=   654.919, eps=       nan
Epoch  45: f=   654.419, eps=       nan
Epoch  46: f=   654.004, eps=       nan
Epoch  47: f=   653.326, eps=       nan
Epoch  48: f=   653.825, eps=       nan
```

```
Epoch  49: f=   652.837, eps=       nan
Epoch  50: f=   654.266, eps=       nan
Epoch  51: f=   652.222, eps=       nan
Epoch  52: f=   652.556, eps=       nan
Epoch  53: f=   651.495, eps=       nan
Epoch  54: f=   652.032, eps=       nan
Epoch  55: f=   651.843, eps=       nan
Epoch  56: f=   651.155, eps=       nan
Epoch  57: f=   652.042, eps=       nan
Epoch  58: f=   651.112, eps=       nan
Epoch  59: f=   650.247, eps=       nan
Epoch  60: f=   649.718, eps=       nan
Epoch  61: f=   649.735, eps=       nan
Epoch  62: f=   649.545, eps=       nan
Epoch  63: f=   650.149, eps=       nan
Epoch  64: f=   649.221, eps=       nan
Epoch  65: f=   648.994, eps=       nan
Epoch  66: f=   648.491, eps=       nan
Epoch  67: f=   649.297, eps=       nan
Epoch  68: f=   648.230, eps=       nan
Epoch  69: f=   647.323, eps=       nan
Epoch  70: f=   648.162, eps=       nan
Epoch  71: f=   647.615, eps=       nan
Epoch  72: f=   648.275, eps=       nan
Epoch  73: f=   647.234, eps=       nan
Epoch  74: f=   647.504, eps=       nan
Epoch  75: f=   646.416, eps=       nan
Epoch  76: f=   646.570, eps=       nan
Epoch  77: f=   646.780, eps=       nan
Epoch  78: f=   646.195, eps=       nan
Epoch  79: f=   645.592, eps=       nan
Epoch  80: f=   647.076, eps=       nan
Epoch  81: f=   645.706, eps=       nan
Epoch  82: f=   645.943, eps=       nan
Epoch  83: f=   645.132, eps=       nan
Epoch  84: f=   646.135, eps=       nan
Epoch  85: f=   645.635, eps=       nan
Epoch  86: f=   644.592, eps=       nan
Epoch  87: f=   645.028, eps=       nan
Epoch  88: f=   643.853, eps=       nan
Epoch  89: f=   644.554, eps=       nan
Epoch  90: f=   643.217, eps=       nan
Epoch  91: f=   643.370, eps=       nan
Epoch  92: f=   643.681, eps=       nan
Epoch  93: f=   643.722, eps=       nan
Epoch  94: f=   643.842, eps=       nan
Epoch  95: f=   643.359, eps=       nan
Epoch  96: f=   643.754, eps=       nan
Epoch  97: f=   642.871, eps=       nan
Epoch  98: f=   643.087, eps=       nan
Epoch  99: f=   642.780, eps=       nan
Result after 100 epochs: f=643.2067260742188
```