EE-UY 4423: Introduction to Machine Learning
Midterm 2, Fall 2016

1. (a) The scatter plot is shown in Fig. 1.

   (b) A simple classifier is to use the boundary $x_2 = 0.5$, shown on the figure. So, we take

   $$z_i = x_2 - 0.5 = \mathbf{w}^\mathsf{T}\mathbf{x} + b,$$

   where $\mathbf{w} = [0, 1]$ and $b = -0.5$.

   (c) We have

   $$P(y_i = 1|\mathbf{x}_i) = \frac{1}{1+e^{z_i}} \Rightarrow P(y_i = 0|\mathbf{x}_i) = 1 - \frac{1}{1+e^{z_i}} = \frac{1}{1+e^{-z_i}}.$$

   Hence, we can write

   $$P(y_i|\mathbf{x}_i) = \frac{1}{1+e^{-u_i}}, \quad u_i = \begin{cases} z_i & \text{if } y_i = 1, \\ -z_i & \text{if } y_i = 0 \end{cases}$$

   Since $1/(1+e^{-u})$ is increasing in u, the likelihood will be minimized for the sample where u_i is the smallest. We calculate u_i for each sample using the following table:

Income (thousands \$), x_{i1}	30	50	70	80	100
Num websites, x_{i2}	0	1	1	2	1
Donate (1=yes or 0=no), y_i	0	1	0	1	1
$z_i = x_{i2} - 0.5$	-0.5	0.5	0.5	1.5	0.5
u_i	0.5	0.5	-0.5	1.5	0.5

   We see u_i is smallest for sample $i = 3$, which is the misclassified point.

   (d) Let z_i' be the new values of the linear discriminant under the new parameters. We have,

   $$z_i' = (\mathbf{w}')^\mathsf{T}\mathbf{x}_i + b' = \alpha\left[\mathbf{w}^\mathsf{T}\mathbf{x}_i + b\right] = \alpha z_i.$$

   Since $\alpha > 0$, the sign of z_i' is the same as z_i. Therefore $\hat{y}_i$ does not change. However, the probabilities do change. Since $\alpha > 1$,

   $$z_i > 0 \Rightarrow z_i' > z_i$$
   $$z_i < 0 \Rightarrow z_i' < z_i.$$

   Hence, for samples where $P(y_i = 1|\mathbf{x}_i) > 0.5$, the probability will increase. For samples where $P(y_i = 1|\mathbf{x}_i) < 0.5$, the probability will decrease.

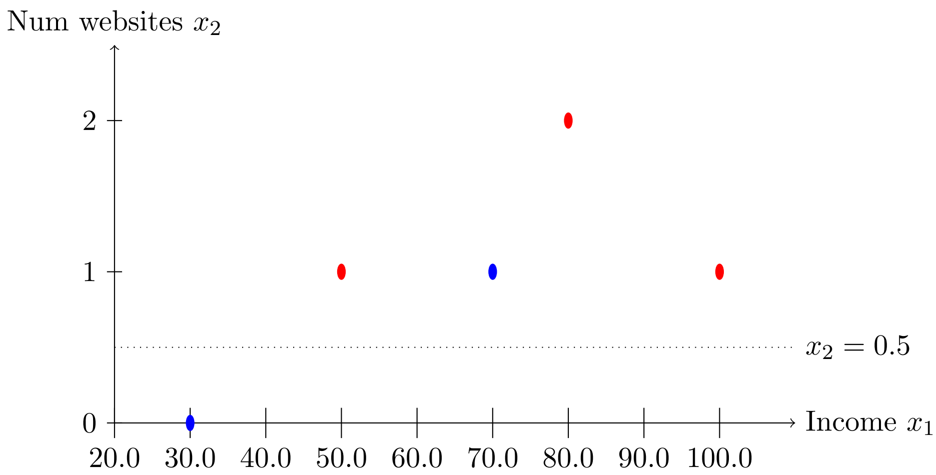

Figure 1: Scatter plot of the data points where the red circles are $y_i = 1$ and blue are $y_i = 0$

(e) You can use the following code:

```python
import numpy as np
def gen_rand(X,w,b):
    z = X.dot(w)+b[:,None]
    p = 1/(1+np.exp(-z))
    n = X.shape[0]
    u = np.random.rand(n)
    y = (u < p)
    return y
```

2. Consider the data set with scalar features x_i and binary class labels $y_i = \pm 1$.

x_i	0	1	3	4	6
y_i	1	-1	1	1	1

A support vector classifier is of the form

$$\hat{y} = \begin{cases} 1 & z > 0 \\ -1 & z < 0, \end{cases} \qquad z = \sum_i \alpha_i y_i K(x_i, x),$$

where $K(x, x')$ is the radial basis function, $K(x, x') = e^{-\gamma(x-x')^2}$, and $\gamma > 0$ is a parameters and the dual coefficients are $\boldsymbol{\alpha} = [1, 1, 1, 0, 0]$

(a) A linear classifier would be of the from

$$\hat{y} = \begin{cases} 1 & z > 0 \\ -1 & z < 0, \end{cases} \qquad z = wx + b,$$

for some weight w and bias b. The classifier would thus separate x into two regions: $\{x > t\}$ and $\{x < t\}$, where $t = b/w$. In this data, for any threshold, there would be at least one point that would be misclassified, so it is not linearly separable.

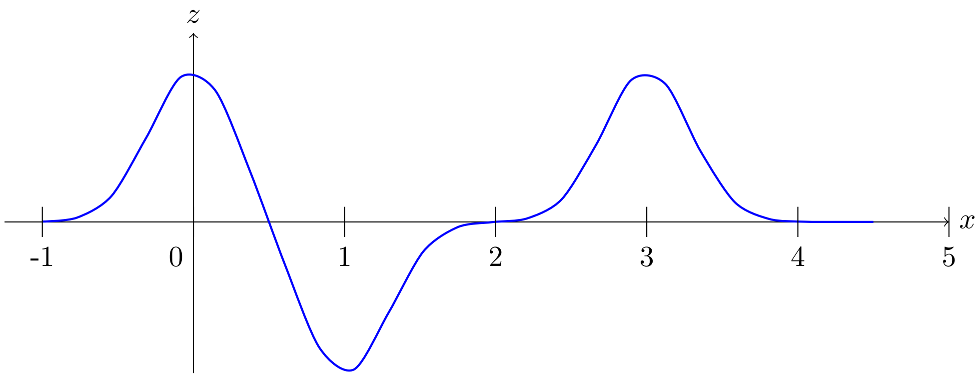

Figure 2: Discriminant z vs. x

(b) The support vectors are the first three samples since they have the non-zero α_i values.

(c) Each function $K(x, x_i)$ is a bell curve centered around x_i. Since γ is large, the width of the bell curve is narrow.

Hint: If γ is large, then

$$e^{-\gamma(x-x_i)^2} \gg e^{-\gamma(x-x_j)^2} \text{ if } |x - x_i| < |x - x_j|.$$

(d) For what values of x is $\hat{y} = 1$?

(e) Complete the following python code to output a vector `yhat` of the predicted values of an SVM classifier at a vector of test values `x`. The SVM uses training data `xtr,ytr`, a radial basis function scale factor `gam` and coefficient `alpha`. Assume that the data is scalar so that `x` and `xtr` are a `numpy` 1D arrays. For full credit, vectorize all operations and write the code without any for loops.

```
import numpy as np
def rbf_predict(xtr,ytr,x,gam,alpha):
    ...
    return yhat
```

Hint: For the most efficient implementation, use python broadcasting to first create a matrix `D` with components `D[i,j]=x[i]−xtr[j]`.

3. An audio engineer wants to design a speech classification system using a neural network. A speaker says one of $K = 10$ words and the system is suppose to determine which word was spoken. For training data, the engineer collects $N = 50000$ samples $(\mathbf{x}_i, y_i)$, $i = 1, \ldots, N$ where each $\mathbf{x}_i$ is a vector of 120 features of the sound and $y_i = 1, \ldots, K$ is the word label. She then tries to fit a neural network of the form,

$$\mathbf{z}_i^{\mathrm{H}} = \mathbf{W}^{\mathrm{H}}\mathbf{x}_i + \mathbf{b}^{\mathrm{H}}, \quad \mathbf{u}_i^{\mathrm{H}} = g_{\mathrm{act}}(\mathbf{z}_i^{\mathrm{H}}),$$
$$\mathbf{z}_i^{\mathrm{O}} = \mathbf{W}^{\mathrm{O}}\mathbf{u}_i^{\mathrm{H}} + \mathbf{b}^{\mathrm{O}}, \quad \hat{y}_i = g_{\mathrm{out}}(\mathbf{z}_i^{\mathrm{O}}),$$

where the activation function is a sigmoid,

$$u_{ij}^{\mathrm{H}} = g_{\mathrm{act}}(z_{ij}^{\mathrm{H}}) = \frac{1}{1 + z_{ij}^{\mathrm{H}}},$$

and the output is the argmax,

$$\hat{y}_i = g_{\mathrm{out}}(\mathbf{z}_i^{\mathrm{O}}) = \arg\max_{k=1,\ldots,K} z_{ik}^{\mathrm{O}}.$$

(a) Suppose the network uses $N_h = 50$ hidden units. What are the dimensions of the parameters $\mathbf{W}^{\mathrm{H}}$, $\mathbf{b}^{\mathrm{H}}$, $\mathbf{W}^{\mathrm{O}}$, $\mathbf{b}^{\mathrm{O}}$.

(b) Given a set of parameters θ, what is a possible loss function $L(\theta)$ that can be used for training the data?

(c) Draw the computation graph showing the mapping from the inputs $\mathbf{x}_i$ and the parameters θ to the loss function, L. Indicate which terms are data and parameters.

(d) Suppose that in backpropagation, you have already computed the derivative, $\partial L/\partial \mathbf{u}_i^{\mathrm{H}}$. Write an expression to compute $\partial L/\partial \mathbf{z}_i^{\mathrm{H}}$. What are the dimensions of $\partial L/\partial \mathbf{u}_i^{\mathrm{H}}$ and $\partial L/\partial \mathbf{z}_i^{\mathrm{H}}$?

(e) Suppose that you have computed $\partial L/\partial \mathbf{z}_i^{\mathrm{H}}$ for all i. Write an expression for the gradients $\nabla_{\mathbf{W}^{\mathrm{H}}} L$ and $\nabla_{\mathbf{b}^{\mathrm{H}}} L$. What are the dimensions of the gradients?