DSC 190 - Homework 02

Due: Wednesday, April 13

Write your solutions to the following problems by either typing them up or handwriting them on another piece of paper. Unless otherwise noted by the problem's instructions, show your work or provide some justification for your answer. Homeworks are due via Gradescope at 11:59 PM.

Problem 1.

Your friend really dislikes Gaussian basis functions, and says "linear basis functions are all you need". You don't agree – you say that linear basis functions won't give you the sort of nice, smooth, non-linear decision boundaries you saw in the lecture on radial basis function networks, no matter how many linear basis functions are used. In fact, you claim that if linear basis functions are used, then the decision boundary in the original space will remain linear.

More formally, let k be the number of linear basis functions used. Any linear basis function takes the form $\varphi(\vec{x}; \vec{a}) = \vec{a} \cdot \vec{x}$, where $\vec{a}$ is a vector of parameters. So define $\varphi_i(\vec{x}; \vec{a}^{(i)}) = \vec{a}^{(i)} \cdot \vec{x}$ to be the ith linear basis function, where $\vec{a}^{(i)}$ is the ith vector in a set of k fixed vectors, $\{\vec{a}^{(1)}, \ldots, \vec{a}^{(k)}\}$.

Show that the prediction function $H(\vec{x}) = w_0 + w_1\varphi_1(\vec{x}) + \ldots + w_k\varphi_k(\vec{x})$ is in fact linear in terms of the coordinates of $\vec{x}$. That is, if $\vec{x} = (x_1, \ldots, x_d)^T$, then H can be written as

$$H(\vec{x}) = w_0 + c_1 x_1 + \ldots c_d x_d,$$

where $c_0, \ldots, c_d$ are constants with respect to $\vec{x}$ (the fact that H can be written in this way implies that the decision boundary is linear in the original space).

Hint: In particular, show that

$$c_j = \sum_{i=1}^{k} w_i \vec{a}_j^{(i)}$$

by plugging in the definition of φ_i to the definition of H above, expanding dot products, and grouping the terms involving each x_i. Here, recall that the notation $\vec{a}_j^{(i)}$ is referring to the jth entry of the vector $\vec{a}^{(i)}$; that is, it is a scalar.

Solution:

Starting from the definition of H:

$$\begin{aligned} H(\vec{x}) &= w_0 + w_1\varphi_1(\vec{x}) + \ldots + w_k\varphi_k(\vec{x}) \\ &= w_0 + \sum_{i=1}^{k} w_i \varphi_i(\vec{x}) \end{aligned}$$

Using the definition of φ_i:

$$= w_0 + \sum_{i=1}^{k} w_i \, \vec{a}^{(i)} \cdot \vec{x}$$

Expanding the dot products:

$$= w_0 + \sum_{i=1}^{k} w_i \left(\vec{a}_1^{(i)} x_1 + \ldots + \vec{a}_d^{(i)} \vec{x}_d \right)$$

We now wish to group all instances of x_1 together, all instances of x_2 together, and so forth. You might see how to do this grouping in one step, but I'll proceed more slowly by first breaking the summation up at the +'s to give summations that only involve one coordinate of x:

$$= w_0 + \sum_{i=1}^{k} w_i \, \vec{a}_1^{(i)} x_1 + \sum_{i=1}^{k} w_i \, \vec{a}_2^{(i)} x_2 + \ldots + \sum_{i=1}^{k} w_i \, \vec{a}_d^{(i)} x_d$$

The x_i's can be factored out of the summations, since they are independent of the index of summation.

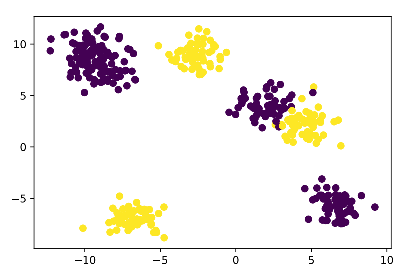
$$\begin{aligned} &= w_0 + x_1 \sum_{i=1}^{k} w_i \vec{a}_1^{(i)} + x_2 \sum_{i=1}^{k} w_i \vec{a}_2^{(i)} + \ldots + x_d \sum_{i=1}^{k} w_i \vec{a}_d^{(i)} \\ &= w_0 + c_1 x_1 + \ldots + c_d x_d \end{aligned}$$

where

$$c_j = \sum_{i=1}^{k} w_i \vec{a}_j^{(i)}$$

Problem 2.

The CSV file located at `https://f000.backblazeb2.com/file/jeldridge-data/001-blobs/data.csv` contains a small data set of points in $\mathbb{R}^2$ along with labels in $\{-1, 1\}$. The first two columns of the CSV are the two features x_1 and x_2, and the third column is the label. When plotted, the data looks like the below:



The purple points belong to the negative class, while the yellow points belong to the positive class.

Train a simple radial basis function network on this data using Gaussian RBFs of the form seen in lecture:

$$\varphi_i(\vec{x}; \vec{\mu}, \sigma) = e^{-\|\vec{x} - \vec{\mu}\|^2 / \sigma^2}.$$

Use 6 centers placed at

$$\begin{aligned}
\vec{\mu}^{(1)} &= (-10, 10)^T \\
\vec{\mu}^{(2)} &= (-3, 10)^T \\
\vec{\mu}^{(3)} &= (-2, 5)^T \\
\vec{\mu}^{(4)} &= (-5, 5)^T \\
\vec{\mu}^{(5)} &= (-6, -5)^T \\
\vec{\mu}^{(6)} &= (-7, -5)^T
\end{aligned}$$

and $\sigma = 4$. Learn the weights $(w_0, \ldots, w_6)$ of the RBF network by minimizing mean square loss in feature space (remember that we saw how to do this using `numpy` in Discussion 01).

This is a coding problem, but it is not autograded. Instead, you'll use your code to answer the questions below. For each part, include the code used to produce the answer. We'll grade your answers based on correctness, and we'll assign partial credit based on your code. You may complete this question in any language of your choosing, though we suggest Python since it is what is used in lecture and discussion.

a) What is the weight vector $\vec{w}$ that was learned by minimizing the mean square loss? Show the code used to train the RBF network, from reading in the data all the way to printing $\vec{w}$.

Solution:

$$\vec{w} = \begin{pmatrix} 0.27 \\ 0.75 \\ 1.16 \\ -2.72 \\ 2.51 \\ -1.43 \\ 0.96 \end{pmatrix}$$

The Python code for finding this weight vector is below:

```
data = np.loadtxt('./data.csv', delimiter=',')

X = data[:,:2]
labels = data[:,-1]

mu = np.array([
    [-10, 10],
    [-3, 10],
    [2, 5],
    [5, 5],
    [6, -5],
    [-7, -5]
])

sigma = 4

def phi(x):
    return np.exp(-np.linalg.norm(x - mu, axis=1)**2 / sigma**2)

Z = np.array([phi(x) for x in X])
```

```python
Z_aug = np.column_stack((
    np.ones(len(Z)),
    Z
))

w = np.linalg.lstsq(Z_aug, labels)[0]
```

b) What is the output of the radial basis function network's prediction function, H, at the point $\vec{x} = (-2, 8)^T$? Your answer should be in the form of a real number don't report the sign of the prediction, but rather the number itself).

Show your code for computing $H(\vec{x})$.

Solution: $H(\vec{x}) \approx 0.59$

```python
def h(x):
    return [1, *phi(x)] @ w
```

c) What is the accuracy of this classifier on the training set?

Solution: 97.5%

```python
np.mean(np.sign(Z_aug @ w) == labels)
```