Machine Learning Worksheet 6

Tomas Ladek, Michael Kratzer
3602673, 3612903
tom.ladek@tum.de, mkratzer@mytum.de

Problem 1

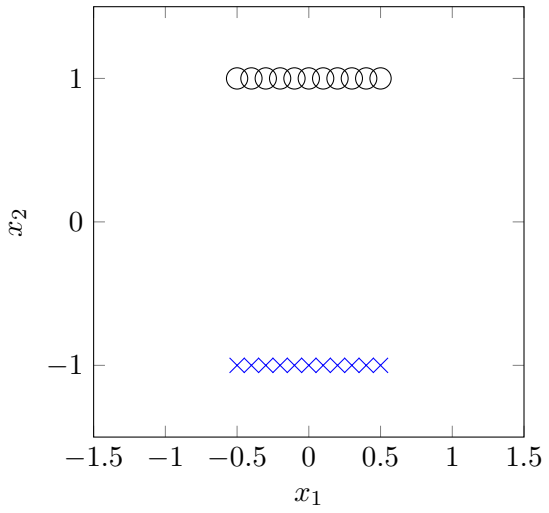
?

Problem 2

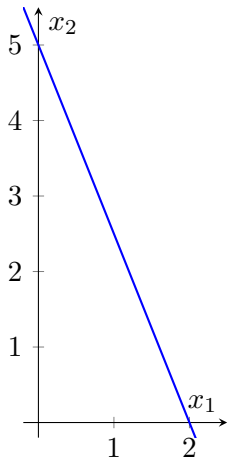
The function

$$\phi(X) : \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \rightarrow \begin{pmatrix} x_1 \cdot x_2 \\ sgn(x_1 \cdot x_2) \end{pmatrix}$$

will transform the data into a space where it will be linearly separable:



Problem 3



General form of this linear classifier:

$$y_k = W^T X + b$$

With W being the normal vector of the two-dimensional hyperplane (line) going through $s_1 = (0, 5)$ and $s_2 = (2, 0)$:

$$W^T = \begin{pmatrix} dx_2 \\ -dx_1 \end{pmatrix}^T \qquad with \quad dx_1 = 0 - 2 = -2 \quad , \; dx_2 = 5 - 0 = 5$$
$$= \begin{pmatrix} 5 \\ 2 \end{pmatrix}^T$$

The bias can be computed by inserting a point on the plane (e.g. s_2) and setting the classifier to 0, like this:

$$b = 0 - W^T X = 0 - \begin{pmatrix} 5 & 2 \end{pmatrix} \begin{pmatrix} 2 \\ 0 \end{pmatrix} = -10$$

The classifier with some possible parameters is therefore:

$$y_k = \begin{pmatrix} 5 \\ 2 \end{pmatrix}^T X - 10$$

Problem 4

```
def pred(X, W, b):
    c = np.add(b, np.dot(X, W.transpose()))

    if np.isscalar(c):
        c = np.array([c])

    for x in np.nditer(c, op_flags=['readwrite']):
            # transform each position with 1 / (1 + exp(-a))
            x[...] = 1 / (1 + np.exp(-1 * x))

    return np.reshape(c, (X.shape[0],1))
```

Problem 5

```
def loglikelihood(X, Z, W, b):
    error = 0
    for i in range(X.shape[0]):
            error += Z[i] * np.log(pred(np.array([X[i]]), W, b))
                  + (1 - Z[i]) * np.log(1 - pred(np.array([X[i]]), W, b))
    return error
```