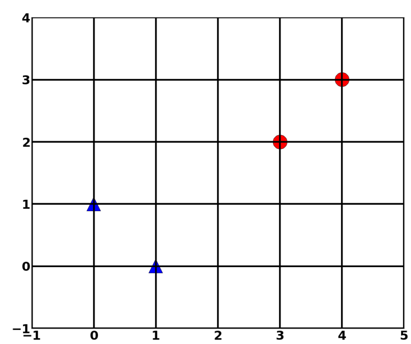
1 Support Vector Machines

Assume we are given dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$, where $x_i \in \mathbb{R}^d$ and $y_i \in \{+1, -1\}$.

In SVM the goal is to find some hyperplane which separates the positive from the negative examples, such that the margin (the minimum distance from the decision boundary to the training points) is maximized. Let the equation for the hyperplane be $\mathbf{w}^\top \mathbf{x} + b = 0$.

(a) You are presented with the following set of data (triangle = +1, circle = -1):



Find the equation (by hand) of the hyperplane $\mathbf{w}^\top \mathbf{x} + b = 0$ that would be used by an SVM classifier. Which points are support vectors?

(b) Let's try to measure the width of the SVM slab (we assume that it was fitted to linearly separable data). We can do this by measuring the distance from one of the support vectors, say x^+, to the plane $w^\top \mathbf{x} + b = 0$. Since the equation of the plane is $\mathbf{w}^\top \mathbf{x} + b = 0$ and since $c(\mathbf{w}^\top \mathbf{x} + b) = 0$ defines the same plane, we have the freedom to choose the normalization of $\mathbf{w}$. Let us choose normalization such that $\mathbf{w}^\top \mathbf{x}_+ + b = +1$ and $\mathbf{w}^\top \mathbf{x}_- + b = -1$, for the positive and negative support vectors respectively. Show that the width of an SVM slab with linearly separable data is $\frac{2}{||\mathbf{w}||}$.

(c) Write SVM as an optimization problem. Conclude that maximizing the margin is equivalent to minimizing $||\mathbf{w}||$. In other words, write the SVM as some $max_{w,b}\frac{2}{||\mathbf{w}||}$ with certain constraints and show that the maximization problem implies $min_{w,b}||w||$ on some constraints.

(d) Will moving points which are not support vectors further away from the decision boundary effect the SVM's hinge loss?

2 Curse of Dimensionality

We have a training set: $(\mathbf{x}^{(1)}, y^{(1)}), \ldots, (\mathbf{x}^{(n)}, y^{(n)})$, where $\mathbf{x}^{(i)} \in \mathbb{R}^d$. To classify a new point $\mathbf{x}$, we can use the nearest neighbor classifier:

$$\text{class}(\mathbf{x}) = y^{(i^*)} \quad \text{where } \mathbf{x}^{(i^*)} \text{ is the nearest neighbor of } \mathbf{x}.$$

Assume any data point $\mathbf{x}$ that we may pick to classify is inside the Euclidean ball of radius 1, i.e. $\|\mathbf{x}\|_2 \leq 1$. To be confident in our prediction, in addition to choosing the class of the nearest neighbor, we want the distance between $\mathbf{x}$ and its nearest neighbor to be small, within some positive ϵ:

$$\|\mathbf{x} - \mathbf{x}^{(i^*)}\|_2 \leq \epsilon \quad \text{for all } \|\mathbf{x}\|_2 \leq 1. \tag{1}$$

What is the minimum number of training points we need for inequality (1) to hold (assuming the training points are well spread)? How does this lower bound depend on the dimension d?

Hint: Think about the volumes of the hyperspheres in d dimensions. The volume of a d dimensional hypersphere is $c(d)r^d$, where r is the radius of the sphere and c is some function that only depends on the dimension of the hypersphere.