Rosenblatt Perceptron

Jarvin Mutatiina (s3555631)
Yulan van Oppen (s2640325)

January 9, 2019

1 Introduction

Many applications of machine learning involve binary classification systems, e.g. the spam filter in your email. The basic processing element in such a system is the linear threshold classifier called the Perceptron. We very briefly introduce some elementary concepts.

A *dichotomy* is a partition of a set into two disjoint subsets. We denote a dichotomy by $\mathfrak{D}_N^P := \{\boldsymbol{\xi}^\mu, S^\mu\}_{\mu=1}^P$, where $\boldsymbol{\xi}_\mu$ is an N-dimensional point and $S^\mu \in \{\pm 1\}$ is the corresponding class label, for $\mu = 1, \dots, P$. The class label as a function of the data points $\boldsymbol{\xi}^\mu$ defines a function $S(\boldsymbol{\xi})$. Suppose this function is *homogeneously linearly separable*, that is, there exists an N-dimensional vector $\mathbf{w}$ such that

$$\text{sign}(\mathbf{w} \cdot \boldsymbol{\xi}^\mu) = S^\mu \quad \text{for} \quad \mu = 1, \dots, P. \tag{1}$$

Should a vector $\mathbf{w}$ satisfying (1) exist, we can assume that any new observations can comfortably fall within the existing partitions. We may re-express (1) by means of *local potentials* $E^\mu := \mathbf{w} \cdot \boldsymbol{\xi}^\mu S^\mu$, as (1) is equivalent to

$$\text{sign}(\mathbf{w} \cdot \boldsymbol{\xi}^\mu S^\mu) = 1, \quad \text{i.e.} \quad E^\mu > 0 \quad \text{for} \quad \mu = 1, \dots, P. \tag{2}$$

Moreover, Rosenblatt states [2] that if the observations are indeed sampled from two linearly separable classes, then a separating decision hyper-plane can be formed between these classes, defined by a vector $\mathbf{w}$ satisfying (1). This hyperplane H is defined as the plane normal to $\mathbf{w}$, through the point $\mathbf{b}$, the *bias*. In other words,

$$H := \left\{ \mathbf{x} \in \mathbb{R}^N \;\middle|\; \mathbf{x} \cdot \mathbf{w} + \mathbf{b} = 0 \right\}.$$

The bias is mainly meant to move the decision boundary away from the origin to cater for correct classification. Note the bias is nonzero only for *inhomogenously separable dichotomies*, that is for dichotomies $\{\boldsymbol{\xi}^\mu, S^\mu\}_{\mu=1}^P$ for which there exists an N-dimensional vector $\mathbf{w}$ and a scalar θ such that

$$\text{sign}(\mathbf{w} \cdot \boldsymbol{\xi}^\mu - \theta) = S^\mu \quad \text{for} \quad \mu = 1, \dots, P.$$

The *perceptron (storage) problem* is to find, for a given dichotomy, a weight vector $\mathbf{w}$ such that (2) holds. With the optimal weight vector, we can correctly classify the inputs into one of the two available classes. In cases of misclassification, the weight vector is adapted until a correct classification is found; as will be described in the next section. We investigate in which circumstances such a weight vector exists.

The remainder of this report is organized as follows. The Methodology section explains how dichotomies are generated, and how we employ the Rosenblatt perceptron algorithm (which is listed in pseudo-code). The Results section describes some graphs that are generated by varying certain parameters. The Discussion section aims to explain these observations.

2 Methodology

The baseline approach is to fix the number of dimensions N, and sample $P = \alpha N$ data points $\boldsymbol{\xi}^\mu$ from a multivariate normal distribution $(\mathbf{0}, \mathbf{I})$, where $\mathbf{I}$ is the N-dimensional identity matrix. The corresponding labels S^μ are independently sampled, where $\mathbb{P}(S^\mu = 1) = \mathbb{P}(S^\mu = -1) = \frac{1}{2}$ for each data point.

Following this, we employ the Rosenblatt Perceptron algorithm 1. The Perceptron Convergence Theorem [2][1] states that Algorithm 1 will find a separating hyperplane for homogeneously linearly separable dichotomies in finitely many steps [1]. In order to illustrate the success rate of the algorithm as a function of α, this is generally done for $\alpha = 0.75, 1, 1.25, \dots, 3$. The success rate is determined by generating n_D dichotomies

randomly for each value of alpha, and by averaging the number of successful runs (denoted by $Q_{l.s}$). Successful runs are those that at some point meet the stopping criterion, before reaching n_{max} epochs (at which the algorithm *always* terminates). This stopping criterion is met when (2) holds for all μ.

Algorithm 1 Rosenblatt perceptron algorithm [3][2]

Input: $\boldsymbol{\xi}, \tau$ ▷ P data points $\boldsymbol{\xi}^\mu \in \mathbb{R}^N$ and corresponding class labels $S^\mu \in \{\pm 1\}$
Output: $\mathbf{w}(T)$ ▷ The final weight vector after T steps

Parameters: N, n_{max}, c ▷ Maximum number of epochs $n_{max} \in \mathbb{N}$ and minimal stability $c > 0$.

Procedure:

1. Initialize $\mathbf{w}(0) = \mathbf{0}$;
2. **For** time steps $t = 1, 2, \ldots, n_{max}$ **do**
   $\mathbf{w}(t) = \mathbf{w}(t-1)$ ▷ Store the per batch evolution of w(t) **For** each data point $\mu = 1, 2, \ldots, P$ **do**
   $E^1 = \mathbf{w}(t) \cdot \boldsymbol{\xi}^1 S^1$
   $\vdots$ ▷ (Re)compute all local potentials E^μ
   $E^P = \mathbf{w}(t) \cdot \boldsymbol{\xi}^P S^P$

   If $E^\mu > c$ for $\mu = 1, 2, \ldots, P$ **then**
   stop ▷ A solution has been found
   else
   $\mathbf{x}(t, \mu) = \frac{1}{N} \boldsymbol{\xi}^\mu S^\mu \cdot \mathbb{1}(E^\mu \leqslant c)$ ▷ Learning step (per sample) using *Hebbian terms* $\boldsymbol{\xi}^\mu S^\mu$
   $\mathbf{w}(t) = \mathbf{w}(t) + \mathbf{x}(t, \mu)$ ▷ Update $\mathbf{w}(t)$
   end if
   end for
   end for

3 Results

In the following section, the number of points P in any given dichotomy will be related to the dimension N of the space under consideration through $P = \alpha N$. We employ the algorithm discussed in the previous section to obtain the fraction of successful runs $Q_{l.s.}$, as a function of α.

Unless specified otherwise, for each figure referred to below, we generate $n_D = 50$ dichotomies $\mathfrak{D}_N^P := \{\boldsymbol{\xi}^\mu, S^\mu\}_{\mu=1}^P$. For each dichotomy we allow for a maximum of $n_{max} = 100$ epochs when using the Rosenblatt algorithm. Moreover, the N-dimensional space we consider has default value $N = 20$. When checking whether the local potentials satisfy $E^\mu > c$, c is by default 0. The figures referred to below contain the y-axis label "fraction of linearly separable dichotomies". It should be remarked, however, that this actually corresponds the the fraction of successful runs of the algorithm. For any dichotomy, more than n_{max} epochs may be needed to find a solution, so some linearly separable dichotomies may be 'labeled' as not linearly separable. The actual fraction is therefore always at least as large as the plotted values.

Holding everything else fixed, varying n_D yields the results depicted in Figure 1. Little variation can be observed. Varying n_{max}, however, significantly changes the results. This is shown in Figure 2. As we vary N, c.f. Figure 3, the graph of $Q_{l.s.}$ as a function of α approaches a step function. The point of discontinuity of the approached graph, however, does not appear to lie on the theoretical value $\alpha = 2$. A reason for this is discussed in the next section.

Enforcing a nonzero minimal stability on the solution, i.e. setting $c > 0$, decreases $Q_{l.s.}$ for any value of α. This is intuitive, as the success criterion becomes more severe (weight vectors satisfying $E^\mu > c > 0$ for each μ also satisfy $E^\mu > 0$ for each μ, but not necessarily vice versa). Figure 4 supports this intuition. Only for these computations, n_D (which was held fixed) was set to 100. This was in an effort to get rid of large oscillations in the graphs of $Q_{l.s.}$ as a function of α, although some persist.

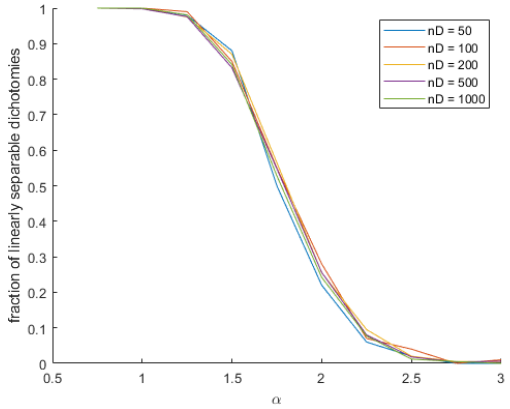

Figure 1: Plots of the fraction of linearly separable dichotomies for several values of n_D. Here $N = 20$, $n_{max} = 100$, $c = 0$, and $\alpha = 0.75, 1, 1.25, \ldots, 3$.

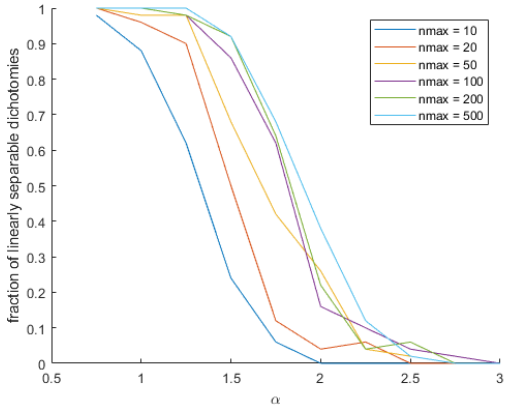

Figure 2: Plots of the fraction of linearly separable dichotomies for several values of n_{max}. Here $N = 20$, $n_D = 50$, $c = 0$ and $\alpha = 0.75, 1, 1.25, \ldots, 3$.

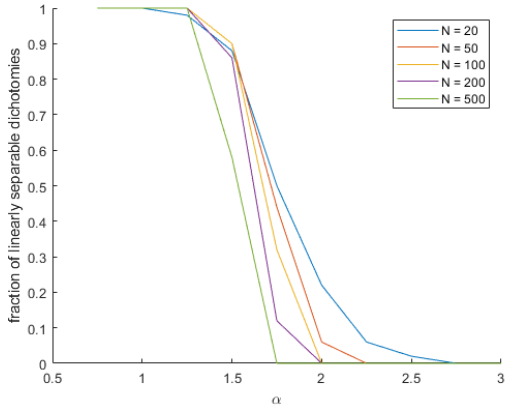

Figure 3: Plots of the fraction of linearly separable dichotomies for several values of N. Here $n_{max} = 100$, $n_D = 50$, $c = 0$, and $\alpha = 0.75, 1, 1.25, \ldots, 3$.

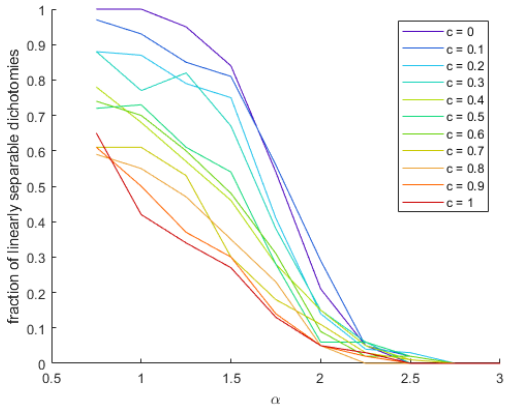

Figure 4: Plots of the fraction of linearly separable dichotomies for several values of c. Here $N = 20$, $n_{max} = 100$ and $n_D = 100$, $c = 0$ and $\alpha = 0.75, 1, 1.25, \ldots, 3$.

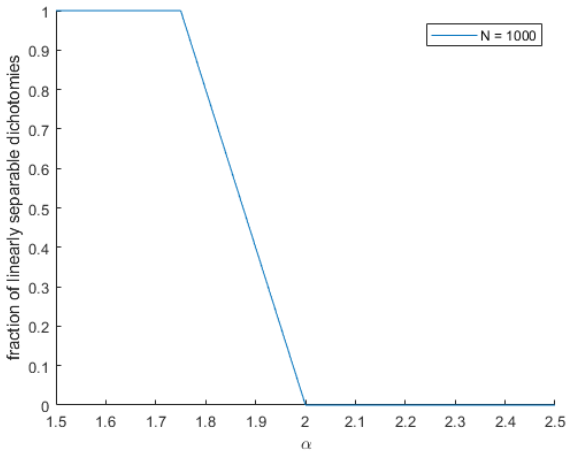

Figure 5: Plots of the fraction of linearly separable dichotomies for $N = 1000$. Here $n_{max} = 1000$, $n_D = 50$, $c = 0$, and $\alpha = 1.5, 1.75, \ldots, 2.5$.

4 Discussion

A higher value of n_D yields results that more accurately depict the true mean of the fraction of successful runs of the Rosenblatt algorithm (since we take the mean of a higher number of samples, each sample being a dichotomy). Figure 1 showed varying n_D had little effect on the graph of $Q_{l.s.}(\alpha)$. This suggests that (for most purposes) $n_D = 50$ considers enough different dichotomies for trustworthy results.

We know from theory that the Rosenblatt algorithm converges for linearly separable dichotomies in finitely many iterations [1], and for $\alpha < 2$ this is the case with probability 1. Finitely many, however, can turn out to be quite a lot. Increasing n_{max} increases the probability of convergence within n_{max} iterations. For $n_{max} = 100$ (our default value), we see the graph of $Q_{l.s.}(\alpha)$ in Figure 2 has generally lower values than its theoretical counterpart. This difference vanishes as n_{max} is sufficiently increased.

This observation moreover explains why the step function approached by the graphs in Figure 3 does not seem to have a discontinuity at $\alpha = 2$, but rather at $\alpha \approx 1.5$. Taking n_{max} considerably larger, i.e $n_{max} = 1000$, shows a low n_{max} really is the cause of this phenomenon. See Figure 5.

In the previous section, we remarked that it was intuitive that $Q_{l.s.}$ decreased when increasing c. This is because for $0 < c_1 < c_2$, a solution with minimal stability c_2 is also a solution with minimal stability c_1. Hence, the probability that the algorithm has a successful run cannot increase when c is increased.

The implementations used may easily be modified to work for inhomogeneously linearly separable dichotomies, using a clamped extra input dimension. The results will most likely not significantly differ from the ones presented in this report, as the only essential difference is a translation of the separating hyperplane. This holds, however, only for data that is not offset from the origin, in which case it might make a difference.

5 Concluding remarks

In conclusion, the perceptron storage problem is solved by finding optimal weights for a given input dichotomy, with which we can correctly classify new inputs. These dichotomies are defined by a separating hyperplane and on occurence of misclassification, the weight vector is adapted by a Hebbian learning term that in turn adjusts the separating hyperplane. This summarizes the *Perceptron algorithm*.

The experiment outcomes are consistent with the conditions associated with the limiting probability $P_{l.s.}$ of linearly separability. That is, for $P = \alpha N$ points of dimension N, as $N \to \infty$, $P_{l.s.} = 1$ if $\alpha \leqslant 2$ and $P_{l.s.} = 0$ otherwise. However, although the Perceptron algorithm converges to a solution for homogeneously linearly separable dichotomies in finitely many steps, the number of steps may still be considerably high. This is supported by the significant change in the experiment outcomes upon increasing n_{max}, allowing for more epochs before termination.

References

[1] Robert.O.Duda et al. *Pattern Classification*. John Wiley Sons, second edition, 2001.

[2] Simon Haykin. *Neural Networks and Learning Machines*. Pearson Prentice Hall, third edition, 2009.

[3] Thomas.M.Cover. Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. *IEEE TRANSACTIONS ON ELECTRONIC COMPUTERS*, pages 326–334.