Logistic Regression and the Perceptron Algorithm

Omar Hijab[*]

Abstract

Hyperplane separability in a two-class dataset is related to the corresponding logistic regression problem. It is shown logistic regression, in its pure un-penalized form, is trainable exactly when the dataset is not separable. This result is contrasted with the behavior of the perceptron algorithm.

Keywords: dataset, separability, logistic regression, perceptron algorithm, binary classifier, decision boundary

1 Introduction

Linear classifiers of two-class datasets are widely used in machine learning. Two approaches to such classifiers are the perceptron algorithm and logistic regression.

Both approaches may be formulated as loss minimization problems, with the goal of achieving an optimal hyperplane

$$m \cdot x + b = 0 \tag{1}$$

separating the dataset's two classes.

In a two-class dataset, each sample has a label $p = 0, 1$. On the other hand, a hyperplane divides the sample space into two half-spaces $q = 0, 1$. When the hyperplane is separating, the p's match the q's for every dataset sample not on the hyperplane.

The perceptron algorithm loss function $J_{pa}(m, b)$ penalizes non-separability of a hyperplane (m, b) directly by measuring the level $y = m \cdot x + b$ of each incorrectly classified sample x. By design, $J_{pa}(m, b) = 0$ exactly when (m, b) is separating.

The logistic regression loss function $J_{lr}(m, b)$ penalizes non-separability of (m, b) by computing a probability $q = q(x)$, depending only on the level y of x, that the sample lies in the class $q = 1$, and measuring the information mismatch between $q(x)$ the label $p = p(x)$. By design, p is never equal to q, and hence $J_{lr}(m, b)$ is never zero.

Given a scale factor $t > 0$, for either problem, rescale $J(m, b)$ to $J(tm, tb)$. This corresponds to rescaling each sample's level y to ty, pushing out the dataset away from the hyperplane (m, b) as $t \to \infty$.

For the perceptron algorithm, $J_{pa}(tm, tb) = tJ_{pa}(m, b)$, for all (m, b). For logistic regression, either $J_{lr}(tm, tb) \to 0$ or $J_{lr}(tm, tb) \to \infty$ as $t \to \infty$, according to whether (m, b) is strictly separating or not separating.

[*] Temple University, Philadelphia, PA, USA, hijab@temple.edu.

Based on this, it is natural to consider $J_{lr}(tm, tb)/t$. Then we have

$$\frac{1}{t} J_{lr}(tm, tb) \to J_{pa}(m, b), \qquad \text{as } t \to \infty, \qquad \text{for all } (m, b). \tag{2}$$

In this sense, the logistic regression loss function is a softened version of the perceptron algorithm loss function. Because this result is not as well-known as it should be, we derive this in §8.

The loss function $J_{pa}(m, b)$ always has a minimizer (m^*, b^*). When the dataset is separable, $\min J_{pa} = 0$, and, when the dataset is not separable, $\min J_{pa} > 0$.

By contrast, existence of a minimizer for $J_{lr}(m, b)$ is connected to the separability of the dataset. When the dataset is not separable, there is a minimizer (m^*, b^*) and $\min J_{lr} > 0$. When the dataset is separable, there is no minimizer. Moreover, when the dataset is strictly separable, the loss is arbitrarily close to zero, $\inf J_{lr} = 0$.

	LR	PA
not separable	$\min J > 0$	$\min J > 0$
separable	no min	$\min J = 0$
strictly separable	$\inf J = 0$	$\min J = 0$

Table 1: Minimizers grid.

Table 1 summarizes the situation.

When the dataset is separable, the well-known perceptron convergence theorem guarantees the convergence of stochastic gradient descent along $J_{pa}(m, b)$ to a separating hyperplane [5]. When the dataset is not separable, the descent sequence remains bounded and thrashes with no convergence [2], [4].

Contrary to what (2) might suggest, for logistic regression the situation is markedly different. When the dataset is not separable, gradient descent along $J_{lr}(m, b)$ converges to a minimizer. When the dataset is separable, the descent sequence diverges to infinity.

	LR	PA
not separable	converges to minimizer	thrashes
separable	diverges to infinity	converges to minimizer

Table 2: Gradient descent grid.

Table 2 summarizes the situation. The LR results are the main results of the paper.

2 Background

Let x_1, x_2, $\ldots$, x_N be the samples of a two-class dataset in sample space, which we assume to be euclidean space $\mathbf{R}^d$ with d features, and let p_1, p_2, $\ldots$, p_N be the sequence of labels reflecting the class membership of the samples. Then the two classes correspond to $p = 0$ and $p = 1$ respectively. Because samples may be repeated, the two classes need not be disjoint.

Let m be a nonzero vector, b a scalar, and $m \cdot x$ the dot product. A *hyperplane* in sample space is specified by (1). When the samples x are scalars, a hyperplane is a point. When each sample has two features, a hyperplane is a line, and, with three features, a hyperplane is a plane.

The *level* of a point x relative to a hyperplane (m, b) is the scalar $y = m \cdot x + b$. Then the hyperplane consists of samples at level zero. If m is a unit vector, the level of x equals the signed distance of x to the hyperplane.

Let $y_k = m \cdot x_k + b$ be the level of the sample x_k relative to a hyperplane, $k = 1, 2, \dots, N$. The hyperplane is *separating* if

$$\begin{aligned} y_k &\le 0, \qquad \text{if } p_k = 0, \\ y_k &\ge 0, \qquad \text{if } p_k = 1, \end{aligned} \qquad k = 1, 2, \dots, N. \tag{3}$$

If the inequalities are strict, the hyperplane is *strictly separating.*

If there is a separating hyperplane, the dataset is *separable.* Otherwise, the dataset is *inseparable.* If there is a strictly separating hyperplane, the dataset is *strictly separable.*

If the dataset lies in a hyperplane, then that hyperplane is separating, so the separability question is only interesting when the dataset does not lie in a hyperplane.

If a dataset is separable, and neither class lies in a separating hyperplane (m, b), any samples in the hyperplane may be considered boundary cases.

In this case, we may select either class and reclassify these samples to belong to that class. This modification does not impact the separability, and the resulting dataset is strictly separable, as can be seen by shifting the bias b slightly. In this sense, separability and strict separability are almost the same.

If a dataset is strictly separable, then the convex hulls K_0 and K_1 of the two classes do not intersect, and there is a shortest line segment connecting them. A *maximum-margin hyperplane* is then the hyperplane orthogonal to such a shortest line segment and passing through its midpoint.

Let μ_0, μ_1 be the means of the two classes, and suppose μ_1 is in a separating hyperplane. Then, relative to the hyperplane, the level of μ_1 is zero, and the sample levels in the class $p = 1$ are nonnegative. Since the level of μ_1 is the average of the sample levels, these sample levels are all zero, hence the class lies in the hyperplane.

Since the same reasoning applies to μ_0, separability and equality of the means imply both means lie in the same separating hyperplane, hence the dataset lies in that hyperplane. Equivalently, if a separable dataset does not lie in a hyperplane, the means of the two classes are distinct.

Given a two-class dataset, a *binary classifier* is a procedure for classifying points in sample space into two classes, in a manner consistent with the dataset. Given a separable dataset, we obtain a binary classifier by selecting a separating hyperplane and assigning points x to classes according to the sign of their level y.

One way to find a separating hyperplane is to minimize a loss function that directly penalizes mismatches. If a hyperplane (m, b) incorrectly classifies a sample x in class $p = 0$, then $y > 0$, and we should lower y. If x is incorrectly classified and in class $p = 1$, then $y < 0$, and we should raise y.

The simplest loss function achieving this is the average of the levels of samples, with each level adjusted by ± according to class, and only taking into account mismatched samples.

Let $\operatorname{relu}(y) = \max(y, 0)$. We are led to the loss function

$$J_{pa}(m, b) = \frac{1}{N} \sum_{k=1}^{N} \left\{ \begin{matrix} \operatorname{relu}(y_k), & \text{if } p_k = 0, \\ \operatorname{relu}(-y_k), & \text{if } p_k = 1, \end{matrix} \right\} \qquad y_k = m \cdot x_k + b. \tag{4}$$

Then $J_{pa}(m,b) = 0$ exactly when (m,b) is separating.

Based on the above heuristics, pushing down this loss function should bring us closer to a separating hyperplane. The *perceptron algorithm* [5] is iteratively following stochastic gradient descent along this loss function. The *perceptron convergence theorem* [5] guarantees this algorithm converges in finitely many steps to a separating hyperplane, provided the dataset is separable.

Given a hyperplane, another binary classifier is obtained by computing a probability q, depending only on the level y of x, that x should be assigned to the class $p = 1$. To be as consistent as possible with the dataset, we choose some measure of discrepancy $I(p,q)$ between probabilities p and q, and we select the hyperplane that minimizes the average J of the discrepancies $I(p_k, q_k)$ between p_k and the probabilities q_k corresponding to the dataset samples x_k, $k = 1, 2, \dots, N$.

Since y is a scalar and q is a probability, we use a squashing function $q = \sigma(y)$ to convert scalars to probabilities.

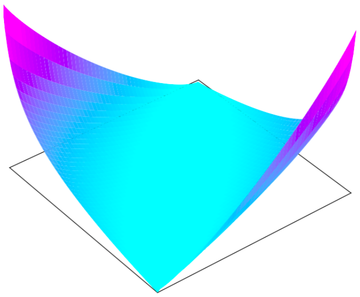

Figure 1: The graph of the relative information over the unit square.

The standard choices are the *relative information* (Figure 1)

$$I(p,q) = p\log\left(\frac{p}{q}\right) + (1-p)\log\left(\frac{1-p}{1-q}\right), \qquad 0 \le p \le 1, 0 < q < 1,$$

and the *sigmoid activation function* (Figure 4)

$$\sigma(y) = \frac{1}{1+e^{-y}}, \qquad -\infty < y < \infty.$$

Since $I \ge 0$ and $I = 0$ only when $p = q$, $I(p,q)$ is a measure of information discrepancy. Because $I(p,q)$ is not symmetric in (p,q), q is thought of as a base probability against which p is compared.

With these choices, *logistic regression* is the minimization of the loss function

$$J_{lr}(m,b) = \frac{1}{N}\sum_{k=1}^{N} I(p_k, q_k), \qquad q_k = \sigma(y_k), \qquad y_k = m \cdot x_k + b,$$

over all m and b.

3 Results

A *minimizer* for a loss function $J(m, b)$ is a weight (m^*, b^*) satisfying $J(m^*, b^*) \leq J(m, b)$ for all (m, b).

Theorem 1. *$J_{pa}(m, b)$ always has a minimizer. When the dataset is separable, $\min J_{pa} = 0$, and, when the dataset is inseparable, $\min J_{pa} > 0$.*

Theorem 2. *Assume the dataset does not lie in a hyperplane. Then $J_{lr}(m, b)$ has at most one minimizer. Moreover the means of the classes agree iff $J_{lr}(m, b)$ has a minimizer with $m^* = 0$.*

Let μ and Q be the mean and variance of the dataset, and let

$$L = \frac{1}{4}\left(1 + |\mu|^2 + \operatorname{trace}(Q)\right). \tag{5}$$

A dataset is *standard* if each feature has mean zero and variance one. If the dataset is standard, $L = (1 + d)/4$.

Theorem 3. *Let (m_1, b_1), (m_2, b_2), (m_3, b_3), ... be a gradient descent sequence for $J_{lr}(m, b)$, with learning rates equal to $1/L$. Then the gradients $\nabla J_{lr}(m_n, b_n)$ converge to zero. Assume neither class lies in a hyperplane. Then either $J_{lr}(m, b)$ has a unique minimizer (m^*, b^*) and the sequence converges to the minimizer,*

$$(m_n, b_n) \to (m^*, b^*), \qquad \textit{as } n \to \infty,$$

or there is no minimizer and the sequence diverges to infinity,

$$|m_n|^2 + b_n^2 \to \infty, \qquad \textit{as } n \to \infty,$$

according to whether the dataset is inseparable or separable.

The details and the proofs of the above results are in §8. By Theorem 2, if the means of the two classes are distinct, the minimizer (m^*, b^*) satisfies $m^* \neq 0$, hence is a hyperplane, the *LR hyperplane*.

4 An Example

A simple example of a two-class logistic regression problem, based on an example in [6], is as follows.

x	p	x	p	x	p	x	p	x	p
0.5	0	.75	0	1.0	0	1.25	0	1.5	0
1.75	0	1.75	1	2.0	0	2.25	1	2.5	0
2.75	1	3.0	0	3.25	1	3.5	0	4.0	1
4.25	1	4.5	1	4.75	1	5.0	1	5.5	1

Figure 2: Months trained and outcomes.

A group of hikers train to scale Mount Rainier. For each hiker, we know the number of months they train, as well as whether or not (p equal 1 or 0) they subsequently succeed at scaling the peak (Figure 2). We use logistic regression to provide a decision boundary or cut-off predicting success.

The samples here are scalars, and the dataset is one-dimensional. In Figure 3, $K_0 \cap K_1$ is the overlap between the two classes. The overlap plays a crucial role in the analysis.

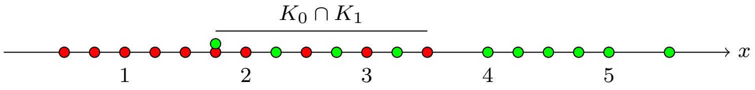

Figure 3: Hikers dataset samples.

Plotting the dataset on the (x, q) plane, the goal is to fit a curve

$$q = \sigma(mx + b) \tag{6}$$

as in Figure 4.

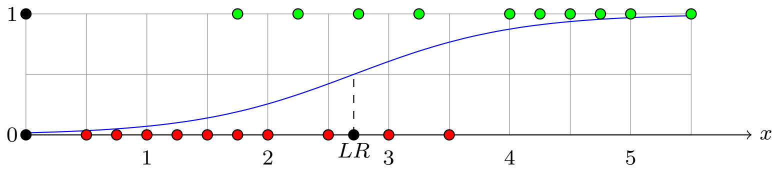

Figure 4: Fitted sigmoid curve for hikers dataset.

The dataset is one-dimensional, so a hyperplane is a point. Neither class lies in a hyperplane, and the dataset is inseparable (Figure 3). Hence $J_{lr}(m, b)$ has a minimizer, and gradient descent is guaranteed to converge to the unique minimizing weight, which turns out to be

$$m^* = 1.50464542, \qquad b^* = -4.0777134.$$

The cut-off $x^* = -b^*/m^* = 2.71008$ is the LR hyperplane (Figure 4).

While this numerical result is in [6], the above framework guaranteeing the existence of (m^*, b^*) is not.

5 Properness

Let $|w|$ denote the absolute value of a scalar weight w or the length of a vector weight w, as the case may be. In our setting, weights are $w = (m, b)$.

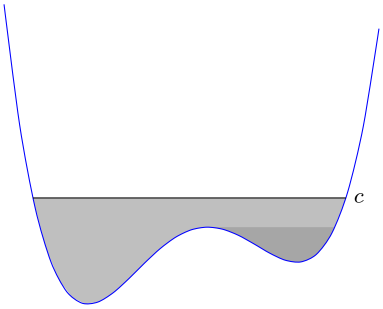

Figure 5: The graph of a proper continuous function.

Let $f(w)$ be a scalar function of weights w, defined on euclidean space. A *minimizer* is a weight w^* satisfying $f(w^*) \leq f(w)$ for all w.

The function $f(w)$ is *proper* if *every sublevel set is bounded:* for every level c, there is a bound C such that

$$f(w) \leq c \qquad \text{implies} \qquad |w| \leq C. \tag{7}$$

A proper function need not be convex everywhere (Figure 5). If the graph of $f(w)$ is the cross-section of a river, then properness means the river never floods its banks, no matter how much it rains. Then the following hold.

- If $f(w)$ is continuous and proper, then $f(w)$ has a minimizer [3, §4.3].
- If $f(w)$ is strictly convex, continuously differentiable, and has a minimizer, then $f(w)$ is proper [3, §4.5].

The parabola $f(w) = w^2$ is strictly convex on the real line, and is proper and has a minimizer. The exponential $f(w) = e^w$ is strictly convex on the real line, but is neither proper nor has a minimizer.

When applying these and the following results, keep in mind $J_{pa}(m, b)$ is continuous but not continuously differentiable, and $J_{lr}(m, b)$ is continuously differentiable.

6 Separability

If x_1, x_2, ..., x_N is a dataset, a *convex combination* of samples is the weighed sum

$$x = t_1 x_1 + t_2 x_2 + \cdots + t_N x_N,$$

with t_k nonnegative, $t_k \geq 0$, and summing to one, $t_1 + t_2 + \cdots + t_N = 1$. The *convex hull* of a dataset is the set of all convex combinations of samples in the dataset.

A *ball with center μ and radius r* is the set B of points x whose distance from μ is no greater than r: $|x - \mu| \leq r$. Let K be any set of points in sample space. Then *K has interior* if there is a ball wholly contained in K. Otherwise, *K has no interior.* With this understood, it is easy to check a hyperplane has no interior.

Let K_0 and K_1 be the convex hulls of the classes of a two-class dataset, and assume neither class lies in a hyperplane. Then wthe following holds [3, §4.5].

- (Hyperplane separation theorem) The dataset is inseparable iff the intersection $K_0 \cap K_1$ has interior. Equivalently, the dataset is separable iff $K_0 \cap K_1$ has no interior.

Any dataset may be "doubled" to a two-class dataset by taking each of the two classes to be the dataset itself. Then, for the doubled dataset, $J_{lr}(m, b) \geq \log 2$ for all (m, b), and $J_{lr}(0, 0) = \log 2$, so $(m^*, b^*) = (0, 0)$ is a minimizer.

The doubled dataset is separable iff the original dataset lies in a hyperplane. Applying the hyperplane separation theorem to the doubled dataset, we conclude a dataset lies in a hyperplane iff its convex hull K has no interior.

7 Gradient Descent

Let $f(w)$ be a scalar function of weights w, defined on euclidean space. A *descent sequence* is a sequence of weights w_1, w_2, w_3, ... satisfying

$$f(w_1) \geq f(w_2) \geq f(w_3) \geq \dots$$

If w is a weight in a sequence of weights, let w^+ denote the next weight in the sequence. A *gradient descent sequence* is a sequence generated by the iteration

$$w^+ = w - t\nabla f(w).$$

The scalar $t > 0$ is the *learning rate.* The learning rates may be constant or may vary along the sequence.

If Q is a symmetric matrix and $u \cdot Qu \leq L$ for all unit vectors u, we write $Q \leq L$. For each w, the second derivative $D^2 f(w)$ is a symmetric matrix.

If the learning rates t of a gradient descent sequence all equal $1/L$ for some L satisfying

$$D^2 f(w) \leq L, \qquad \text{for all } w, \tag{8}$$

the gradient descent sequence is *short-step.*

When the gradient descent sequence is short-step, $t = 1/L$,

$$f(w^+) \leq f(w) - \frac{t}{2}|\nabla f(w)|^2 \tag{9}$$

[3, §7.3], hence a short-step gradient descent sequence is a descent sequence.

If $f(w)$ is bounded below, the limit of the loss function along a descent sequence exists. From (9), we conclude: Along a short-step gradient descent sequence, the limit of the gradient of the loss function is zero.

A sequence *subconverges* to w^* if some subsequence converges to w^*. If a short-step gradient descent sequence subconverges to w^*, and $\nabla f(w)$ is continuous, then $\nabla f(w^*) = 0$.

For a convex function, $\nabla f(w^*) = 0$ iff w^* is a minimizer. We conclude w^* is a minimizer if moreover $f(w)$ is convex.

8 Proofs of Results

We delay the proof of Theorem 1 till the end, and we focus first on $J_{lr}(m, b)$, which we denote more simply as $J(m, b)$ in the derivation of Theorems 2 and 3 below. We start by computing the derivatives of $J(m, b)$.

The *cumulant-generating function* of a fair coin, ignoring a constant term, is

$$Z(y) = \log(1 + e^y). \tag{10}$$

Then

$$Z'(y) = q, \qquad Z''(y) = q' = q(1-q), \qquad q = \sigma(y).$$

Let $I(p)$ be the *absolute information,*

$$I(p) = p \log p + (1-p)\log(1-p), \qquad 0 \le p \le 1.$$

Then

$$I(p,q) = I(p) - py + Z(y), \qquad q = \sigma(y), \tag{11}$$

This last identity, the *information error identity,* implies $I(p)$ and $Z(y)$ are dual convex functions [3, §4.1].

Since $I(p) = 0$ when $p = 0, 1$, (11) implies

$$I(p,q) = \begin{cases} Z(y), & \text{if } p = 0, \\ Z(y) - y = Z(-y), & \text{if } p = 1, \end{cases} \qquad q = \sigma(y). \tag{12}$$

Consequently, $J(m, b)$ is the standard "cross-entropy" loss function.

From (11), we have

$$\frac{d}{dy} I(p,q) = q - p, \qquad \frac{d^2}{dy^2} I(p,q) = q(1-q), \qquad q = \sigma(y).$$

By the chain rule,

$$\frac{d}{dt} I(p, \sigma(y+tv)) = (q-p)v, \qquad \frac{d^2}{dt^2} I(p, \sigma(y+tv)) = q(1-q)v^2.$$

Let v_0 be a vector and v_1 a scalar, and let $v = v_0 \cdot x + v_1$. Then

$$(m + tv_0) \cdot x + (b + tv_1) = (m \cdot x + b) + t(v_0 \cdot x + v_1) = y + tv.$$

With

$$q_k = \sigma(y_k) = \sigma(m \cdot x_k + b), \qquad k = 1, 2, \dots, N,$$

the first directional derivative of the loss function is

$$\left.\frac{d}{dt}\right|_{t=0} J(m + tv_0, b + tv_1) = \frac{1}{N} \sum_{k=1}^{N} (q_k - p_k)(v_0 \cdot x_k + v_1), \tag{13}$$

and the second directional derivative of the loss function is

$$\left.\frac{d^2}{dt^2}\right|_{t=0} J(m + tv_0, b + tv_1) = \frac{1}{N} \sum_{k=1}^{N} q_k(1-q_k)(v_0 \cdot x_k + v_1)^2. \tag{14}$$

Since $0 < q_k < 1$, $k = 1, 2, \dots, N$, by (14), the second directional derivative is nonnegative, so $J(m, b)$ is convex.

If the second directional derivative equals zero for some m, b, v_0, and v_1, by (14), the dataset satisfies $v_0 \cdot x_k + v_1 = 0$, $k = 1, 2, \dots, N$. Thus the dataset lies in a hyperplane, unless $v_0 = v_1 = 0$.

Under the assumptions of Theorem 2, the dataset does not lie in a hyperplane, hence $v_0 = v_1 = 0$. This establishes the strict convexity of $J(m, b)$. Since a strictly convex function has at most one minimizer, this establishes the first portion of Theorem 2.

Since (13) shows the gradient of the loss function is a continuous function of (m, b), we see existence of a minimizer and properness of the loss function are equivalent, when the dataset does not lie in a hyperplane.

A point (m, b) is a *critical point* if the first directional derivative (13) vanishes in all directions (v_0, v_1). For any convex function, a point is critical iff it is a minimizer.

Let n be the number of samples in the class $p = 1$, and let μ_0, μ_1 be the means of the classes $p = 0$, $p = 1$. Then

$$\frac{n}{N}\mu_1 = \frac{1}{N}\sum_{p_k=1} x_k, \qquad \frac{N-n}{N}\mu_0 = \frac{1}{N}\sum_{p_k=0} x_k.$$

Given b, let $q = \sigma(b)$. If $(0, b)$ is a critical point, then the first directional derivative vanishes, and (13) implies

$$\frac{N-n}{N}\, q\,(\mu_0 \cdot v_0 + v_1) = \frac{n}{N}\,(1-q)\,(\mu_1 \cdot v_0 + v_1) \tag{15}$$

for all v_0 and v_1. Taking $v_1 = 1$ and $v_0 = 0$ in (15) implies $q = n/N$. Using this, and taking $v_1 = 0$ and v_0 arbitrary in (15), we conclude $\mu_0 = \mu_1$.

Conversely, if $\mu_0 = \mu_1$, let $q = n/N$ be the proportion of samples satisfying $p_k = 1$, and let $b = \sigma^{-1}(q)$. Then (15) holds for all v_0 and v_1. It follows (13) vanishes with $(m, b) = (0, b)$, hence $(0, b)$ is a critical point. This establishes the second portion of Theorem 2.

By (12),

$$I(p, q) \le \log 2, \qquad q = \sigma(y), \qquad \begin{cases} \text{if } p = 0 \text{ and } y \le 0, \\ \text{if } p = 1 \text{ and } y \ge 0. \end{cases} \tag{16}$$

For Theorem 3, assume the dataset is separable. Then there is a separating hyperplane (m, b). By (16), $I(p_k, q_k) \le \log 2$ for $k = 1, 2, \dots, N$, hence $J(m, b) \le \log 2$. But (tm, tb) is the same hyperplane, so

$$J(tm, tb) \le \log 2, \qquad t > 0.$$

Since this contradicts (7), $J(m, b)$ is not proper, hence there is no minimizer.

On the other hand, suppose the dataset is inseparable. Let K_0 and K_1 be the convex hulls of the two classes. If neither class lies in a hyperplane, by the hyperplane separation theorem, the intersection $K_0 \cap K_1$ has interior, so there is a ball B in $K_0 \cap K_1$.

Let μ and r be the center and radius of B. We establish properness by showing

$$J(m, b) \le c \qquad \text{implies} \qquad |m| + |b| \le \frac{cN}{r} \cdot (1 + r + |\mu|)\,. \tag{17}$$

Since $w = (m, b)$ implies

$$|w| = \sqrt{|m|^2 + b^2} \le |m| + |b|,$$

(17) implies properness.

If $J(m, b) \le c$, then $I(p_k, q_k) \le cN$, $k = 1, 2, \dots, N$. Since $y < Z(y)$, by (12),

$$\begin{aligned} y_k < Z(y_k) = I(p_k, q_k) \le cN, &\qquad \text{if } p_k = 0, \\ -y_k < Z(-y_k) = I(p_k, q_k) \le cN, &\qquad \text{if } p_k = 1, \end{aligned} \qquad k = 1, 2, \dots, N.$$

By taking convex combinations,

$$\begin{aligned} y &\leq cN, && \text{for } x \text{ in } K_0, \\ -y &\leq cN, && \text{for } x \text{ in } K_1. \end{aligned}$$

From this,

$$|m \cdot x + b| \leq cN, \qquad \text{for } x \text{ in } K_0 \cap K_1.$$

If v is a unit vector, the points $x_\pm = \mu \pm rv$ are in B. Since

$$2rm \cdot v = (m \cdot x_+ + b) - (m \cdot x_- + b),$$

we have

$$2r|m \cdot v| \leq |m \cdot x_+ + b| + |m \cdot x_- + b| \leq 2cN.$$

Choosing $v = m/|m|$, we obtain

$$|m| \leq \frac{cN}{r}.$$

The point μ is in B. Since

$$b = (m \cdot \mu + b) - m \cdot \mu,$$

we have

$$|b| \leq |m \cdot \mu + b| + |m \cdot \mu| \leq cN + |m|\,|\mu|.$$

This leads to (17), establishing properness, hence existence of a minimizer of $J(m, b)$.

Let

$$\mu = \frac{1}{N}\sum_{k=1}^N x_k, \qquad Q = \frac{1}{N}\left(\sum_{k=1}^N x_k \otimes x_k\right) - \mu \otimes \mu$$

be the mean and variance of the dataset. Then, with L given by (5),

$$4L = 1 + |\mu|^2 + \operatorname{trace}(Q) = \frac{1}{N}\sum_{k=1}^N \left(1 + |x_k|^2\right).$$

Since $q(1-q) \leq 1/4$ and

$$(v_0 \cdot x + v_1)^2 \leq \left(|x|^2 + 1\right)\left(|v_0|^2 + v_1^2\right),$$

by (14), we conclude

$$D^2 J(m, b) \leq L, \qquad \text{for all } (m, b). \tag{18}$$

Let w_1, w_2, w_3, ... be a gradient descent sequence for the loss function, with L given by (5). Then, by (18), the sequence is short-step. Hence, by (9), the sequence is a descent sequence. It follows the sequence remains in a sublevel set.

Since $J(m, b)$ is strictly convex, if the sequence subconverges to (m^*, b^*), then (m^*, b^*) is the unique minimizer.

If the dataset is inseparable, there is a minimizer, hence $J(m, b)$ is proper, hence sublevel sets are bounded, hence the sequence remains bounded. If the sequence subconverges to some (m^*, b^*), then (m^*, b^*) is the unique minimizer, and the sequence converges to (m^*, b^*).

On the other hand, if the dataset is separable, there is no minimizer, hence the sequence cannot subconverge to any (m^*, b^*). Thus the sequence diverges to infinity, completing the proof of Theorem 3.

Theorem 3 and its proof remain valid for non-constant learning rates t, as long as $\epsilon \le t \le 1/L$, for some fixed $\epsilon > 0$.

From Figure 1, $I(p,q) = 0$ at two of the corners of the unit square, and $I(p,q) = \infty$ at the other two corners. Let x be a sample not on a hyperplane (m,b). With $y = m \cdot x + b$, it follows $I(p, \sigma(ty))$ goes to zero or to infinity, as $t \to \infty$, according to whether (m,b) correctly classifies x or not. We conclude $J(tm,tb) \to \infty$ as $t \to \infty$, if (m,b) is not separating, and $J(tm,tb) \to 0$ as $t \to \infty$, if (m,b) is strictly separating.

Let

$$\mathbf{1}(y > 0) = \begin{cases} 0, & y < 0, \\ \frac{1}{2}, & y = 0, \\ 1, & y > 0, \end{cases}$$

be the step function and let $\operatorname{relu}(y) = \max(y, 0)$. Then $\sigma(ty) \to \mathbf{1}(y > 0)$ as $t \to \infty$, and

$$\operatorname{relu}(y) = y\mathbf{1}(y > 0) \qquad \text{and} \qquad \operatorname{relu}(y) - y = \operatorname{relu}(-y). \tag{19}$$

For (2), by l'Hopital's rule,

$$\lim_{t\to\infty} \frac{1}{t} Z(ty) = \lim_{t\to\infty} Z'(ty)y = \lim_{t\to\infty} \sigma(ty)y = \operatorname{relu}(y). \tag{20}$$

This shows $Z(y)$ is a softened version of $\operatorname{relu}(y)$. Then (11), (19), and (20) imply

$$\lim_{t\to\infty} \frac{1}{t} I(p, \sigma(ty)) = \begin{cases} \operatorname{relu}(y) & p = 0, \\ \operatorname{relu}(-y), & p = 1. \end{cases}$$

This last equation implies (2).

Finally, for Theorem 1, $J_{pa}(m,b) = 0$ when (m,b) is separating. To show $J_{pa}(m,b)$ has a minimizer when the dataset is inseparable, it is enough to verify inseparability implies properness of $J_{pa}(m,b)$.

Since

$$\max(\operatorname{relu}(y), \operatorname{relu}(-y)) = |y|,$$

the proof of (17), suitably modified, remains valid for $J(m,b) = J_{pa}(m,b)$.

9 Discussion

The takeaway is we have two mutually exclusive cases. Either a dataset is separable or not. If a dataset is separable, *PA* gradient descent converges to a decision boundary. If a dataset is inseparable, *LR* gradient descent converges to a decision boundary.

In practice, referring to *PA*, [1] states

> *... the number of steps required to achieve convergence could still be substantial, and in practice, until convergence is achieved, we will not be able to distinguish between a nonseparable problem and one that is simply slow to converge ...*

Theorem 3 suggests using *LR* to check for separability. When following short-step gradient descent, we know the gradients converge to zero in all cases. However, in practice, there seems to be a substantial difference in the rate of convergence, according to whether or not the dataset is separable.

10 Terminology

Because there is some terminology confusion in the literature, we go over the usage of some of the terms used above.

1. In the literature, $I(p,q)$ is called the "Kullback-Liebler divergence" or "relative entropy". We prefer the term relative information because this terminology is more descriptive and consistent with the absolute information $I(p)$, and because $I(p,q)$ is convex.
2. In Python, as of this writing, `scipy.stats.entropy(p)` returns the *absolute entropy* $H(p) = -I(p)$, and `scipy.stats.entropy(p,q)` returns the relative information $I(p,q)$, not the *relative entropy* $H(p,q) = -I(p,q)$. Apart from being incorrect, this is inconsistent, even within Python.
3. In the literature, $Z(y) - p \cdot y$ is called the "cross-entropy". We prefer the term *cross-information* because this terminology is consistent with the terminology for $I(p)$ and $I(p,q)$, and because (see (11)) the cross-information equals $I(p,q) - I(p)$ when $q = \sigma(y)$.
4. To further support our terminology choices, we note entropy is the negative of information, entropy is concave, information is convex, and loss functions are minimized, not maximized. Of course, the issue here is terminology, not the choice of loss function: the loss functions in the literature are identical with the loss functions here.

11 Disclosure of Funding

This research was not supported by third-party funding. Moreover, the author reports there are no competing interests to declare.

References

[1] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[2] H. D. Block and S. A. Levin. "On the boundedness of an iterative procedure for solving a system of linear inequalities". In: *Proceedings of the American Mathematical Society* 26.2 (1970), pp. 229–235.

[3] O. Hijab. *Math for Data Science*. Springer, 2025.

[4] A. J. Novikov. *On convergence proofs for perceptrons*. Tech. rep. Office of Naval Research, 1963.

[5] F. Rosenblatt. *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms*. Spartan Books, 1962.

[6] Wikipedia. *Logistic Regression*. URL: https://en.wikipedia.org/wiki/Logistic_regression.