Classification tasks, the Perceptron model

Chris Cornwell

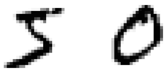
October 21, 2025

Outline

Outline

Example of Classification

Use some model to determine a digit that was (hand)written in an image



$\rightsquigarrow$ `0`, `1`, `2`, `3`, `4`, `5`, `6`, `7`, `8`, or `9`.

Example of Classification

Use some model to determine a digit that was (hand)written in an image

$\rightsquigarrow$ `0`, `1`, `2`, `3`, `4`, `5`, `6`, `7`, `8`, or `9`.

- Convert image to a vector (*in some way*) $\longrightarrow$ $\mathbf{x}$.
- Your model's output: the (predicted) digit.

Example of Classification

Use some model to determine a digit that was (hand)written in an image

$\rightsquigarrow$ `0`, `1`, `2`, `3`, `4`, `5`, `6`, `7`, `8`, or `9`.

- Convert image to a vector (*in some way*) $\rightarrow$ $\mathbf{x}$.
- Your model's output: the (predicted) digit.

In data provided, $\{(\mathbf{x}_i, y_i)\}$, observed ("correct") label is $y_i \in \{0, 1, \ldots, 9\}$.

Example of Classification

Use some model to determine a digit that was (hand)written in an image

$\rightsquigarrow$ 0, 1, 2, 3, 4, 5, 6, 7, 8, or 9.

- Convert image to a vector (*in some way*) $\rightarrow$ $\mathbf{x}$.
- Your model's output: the (predicted) digit.

In data provided, $\{(\mathbf{x}_i, y_i)\}$, observed ("correct") label is $y_i \in \{0, 1, \ldots, 9\}$.
Value of y is on number line; but, consider it a label (or, one of a few separate "buckets") used to organize different points $\mathbf{x}$. (When $y_i = 5$, predicting 4 is not any better than 0.)

Close only counts in ~~horseshoes~~ ...Regression

In linear regression, on indpt. variables $x_1, x_2, \ldots, x_N$, had (affine) linear function $y \approx b + w_1 x_1 + w_2 x_2 + \ldots + w_N x_N$;
values of function $\leftrightarrow$ prediction $\hat{y}$; error term $\varepsilon = y - \hat{y}$.

[1]Should consider the output y here to be a random variable, with distribution that depends on $\mathbf{x}$.

Close only counts in ~~horseshoes~~ ...Regression

In linear regression, on indpt. variables $x_1, x_2, \ldots, x_N$, had (affine) linear function $y \approx b + w_1 x_1 + w_2 x_2 + \ldots + w_N x_N$;
values of function $\leftrightarrow$ prediction $\hat{y}$; error term $\varepsilon = y - \hat{y}$.
In regression, the **linear model** $\mathbf{x} \mapsto \hat{y}$ approximates the relationship $\mathbf{x} \mapsto y$.[1] We expect that $|y - \hat{y}|$ is <u>almost never</u> exactly 0; a good model: one where $|y - \hat{y}|$ is small on average (but, still positive).

[1]Should consider the output y here to be a random variable, with distribution that depends on $\mathbf{x}$.

Close only counts in ~~horseshoes~~ …Regression

In linear regression, on indpt. variables $x_1, x_2, \ldots, x_N$, had (affine) linear function $y \approx b + w_1 x_1 + w_2 x_2 + \ldots + w_N x_N$;
values of function $\leftrightarrow$ prediction $\hat{y}$; error term $\varepsilon = y - \hat{y}$.
In regression, the **linear model** $\mathbf{x} \mapsto \hat{y}$ approximates the relationship $\mathbf{x} \mapsto y$.[1] We expect that $|y - \hat{y}|$ is <u>almost never</u> exactly 0; a good model: one where $|y - \hat{y}|$ is small on average (but, still positive).
"Classification" tasks: the value y is a label, might not even be a number. The prediction $\hat{y}$ is either wrong, or not wrong; close doesn't count. Good model: when $\hat{y} = y$ as often as possible.

[1] Should consider the output y here to be a random variable, with distribution that depends on $\mathbf{x}$.

Outline

Classification tasks

Perceptron model

Perceptron algorithm

A linear model for classification

Binary classification: Data from $\mathbb{R}^N$ for some $N > 0$ and only two labels, $\{1, -1\}$.

[2] Notation here is that $x_1, \ldots, x_N$ are the coordinates of the vector $\mathbf{x}$.

A linear model for classification

Binary classification: Data from $\mathbb{R}^N$ for some $N > 0$ and only two labels, $\{1, -1\}$.

A hyperplane in $\mathbb{R}^N$ is an (affine) linear subspace that separates $\mathbb{R}^N$ in two. Given numbers $w_1, w_2, \ldots, w_d$, and b, it can be thought of as the set of points $\mathbf{x} \in \mathbb{R}^N$ where the linear function $y = b + w_1 x_1 + \ldots + w_N x_N$ has value zero[2]:

$$\left\{ \left(x_1, \ldots, x_N \right) \ : \ w_1 x_1 + w_2 x_2 \ldots + w_d x_d + b = 0 \right\}.$$

[2] Notation here is that $x_1, \ldots, x_N$ are the coordinates of the vector $\mathbf{x}$.

A linear model for classification

Binary classification: Data from $\mathbb{R}^N$ for some $N > 0$ and only two labels, $\{1, -1\}$.

A hyperplane in $\mathbb{R}^N$ is an (affine) linear subspace that separates $\mathbb{R}^N$ in two. Given numbers $w_1, w_2, \ldots, w_d$, and b, it can be thought of as the set of points $\mathbf{x} \in \mathbb{R}^N$ where the linear function $y = b + w_1x_1 + \ldots + w_Nx_N$ has value zero[2]:

$$\{(x_1, \ldots, x_N) \ : \ w_1x_1 + w_2x_2 \ldots + w_dx_d + b = 0\}.$$

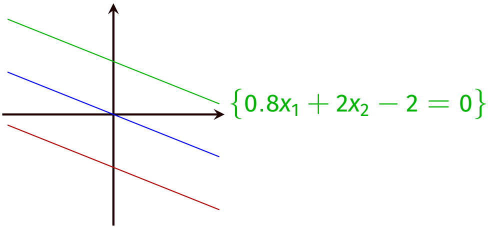


Figure: A few hyperplanes in $\mathbb{R}^2$.

[2] Notation here is that $x_1, \ldots, x_N$ are the coordinates of the vector $\mathbf{x}$.

A linear model for classification

Binary classification: Data is from $\mathbb{R}^N$ for some $N > 0$ and we only have two labels, $\{1, -1\}$.

A hyperplane in $\mathbb{R}^N$ is an (affine) linear subspace that separates $\mathbb{R}^N$ in two. Given numbers $w_1, w_2, \ldots, w_d$, and b, it can be thought of as the set of points $\mathbf{x} \in \mathbb{R}^N$ where the linear function $y = b + w_1x_1 + \ldots + w_Nx_N$ has value zero[3]:

$$\left\{ \left(x_1, \ldots, x_N\right) \ : \ w_1x_1 + w_2x_2 \ldots + w_dx_d + b = 0 \right\}.$$

- Calling the hyperplane H and rewriting this in vector form: if $\mathbf{w} = \left(w_1, w_2, \ldots, w_N\right)$ and $\tilde{\mathbf{w}} = \left(b, w_1, \ldots, w_N\right)$, then H is the set of $\mathbf{x}$ so that $\tilde{\mathbf{x}}^\top \tilde{\mathbf{w}} = \mathbf{w} \cdot \mathbf{x} + b = 0$.

[3]Notation here is that $x_1, \ldots, x_N$ are the coordinates of the vector $\mathbf{x}$.

A linear model for classification

Binary classification: Data is from $\mathbb{R}^N$ for some $N > 0$ and we only have two labels, $\{1, -1\}$.
A hyperplane in $\mathbb{R}^N$ is an (affine) linear subspace that separates $\mathbb{R}^N$ in two. Given numbers $w_1, w_2, \ldots, w_d$, and b, it can be thought of as the set of points $\mathbf{x} \in \mathbb{R}^N$ where the linear function $y = b + w_1x_1 + \ldots + w_Nx_N$ has value zero[3]:

$$\big\{(x_1, \ldots, x_N)\ :\ w_1x_1 + w_2x_2 \ldots + w_dx_d + b = 0\big\}.$$

- Calling the hyperplane H and rewriting this in vector form: if $\mathbf{w} = (w_1, w_2, \ldots, w_N)$ and $\tilde{\mathbf{w}} = (b, w_1, \ldots, w_N)$, then H is the set of $\mathbf{x}$ so that $\tilde{\mathbf{x}}^\top \tilde{\mathbf{w}} = \mathbf{w} \cdot \mathbf{x} + b = 0$.
- H separates $\mathbb{R}^N$ into two parts: those $\mathbf{x}$ where $\mathbf{w} \cdot \mathbf{x} + b$ is positive and those where $\mathbf{w} \cdot \mathbf{x} + b$ is negative.

[3]Notation here is that $x_1, \ldots, x_N$ are the coordinates of the vector $\mathbf{x}$.

A linear model for classification

Binary classification: Data is from $\mathbb{R}^N$ for some $N > 0$ and we only have two labels, $\{1, -1\}$.
A hyperplane in $\mathbb{R}^N$ is an (affine) linear subspace that separates $\mathbb{R}^N$ in two. Given numbers $w_1, w_2, \ldots, w_d$, and b, it can be thought of as the set of points $\mathbf{x} \in \mathbb{R}^N$ where the linear function $y = b + w_1 x_1 + \ldots + w_N x_N$ has value zero[3]:

$$\{(x_1, \ldots, x_N) \ : \ w_1 x_1 + w_2 x_2 \ldots + w_d x_d + b = 0\}.$$

- Calling the hyperplane H and rewriting this in vector form: if $\mathbf{w} = (w_1, w_2, \ldots, w_N)$ and $\tilde{\mathbf{w}} = (b, w_1, \ldots, w_N)$, then H is the set of $\mathbf{x}$ so that $\tilde{\mathbf{x}}^\top \tilde{\mathbf{w}} = \mathbf{w} \cdot \mathbf{x} + b = 0$.
- H separates $\mathbb{R}^N$ into two parts: those $\mathbf{x}$ where $\mathbf{w} \cdot \mathbf{x} + b$ is positive and those where $\mathbf{w} \cdot \mathbf{x} + b$ is negative.
- $\mathbf{w}$ is a vector that is orthogonal to H (which is $(N-1)$-dimensional); $|b|$ and $\|\mathbf{w}\|$ relate to how far H is translated away from the origin.

[3] Notation here is that $x_1, \ldots, x_N$ are the coordinates of the vector $\mathbf{x}$.

Half-space model

Using the notation from last slide: a half-space model in $\mathbb{R}^N$ is determined by $\tilde{\mathbf{w}} = (b, w_1, w_2, \ldots, w_N)$, with a corresponding hyperplane H.

Half-space model

Using the notation from last slide: a half-space model in $\mathbb{R}^N$ is determined by $\tilde{\mathbf{w}} = (b, w_1, w_2, \ldots, w_N)$, with a corresponding hyperplane H.
Given $\mathbf{x} \in \mathbb{R}^N$, the half-space model can be though of as a function: $h : \mathbb{R}^d \setminus H \to \{1, -1\}$, with

Half-space model

Using the notation from last slide: a <u>half-space model</u> in $\mathbb{R}^N$ is determined by $\tilde{\mathbf{w}} = (b, w_1, w_2, \dots, w_N)$, with a corresponding hyperplane H.
Given $\mathbf{x} \in \mathbb{R}^N$, the half-space model can be though of as a function: $h : \mathbb{R}^d \setminus H \to \{1, -1\}$, with

- (Positive side) set $h(\mathbf{x}) = 1$ if $\mathbf{w} \cdot \mathbf{x} + b > 0$.
- (Negative side) set $h(\mathbf{x}) = -1$ if $\mathbf{w} \cdot \mathbf{x} + b < 0$.

Half-space model

Using the notation from last slide: a half-space model in $\mathbb{R}^N$ is determined by $\tilde{\mathbf{w}} = (b, w_1, w_2, \ldots, w_N)$, with a corresponding hyperplane H.
Given $\mathbf{x} \in \mathbb{R}^N$, the half-space model can be though of as a function: $h : \mathbb{R}^d \setminus H \to \{1, -1\}$, with

- (Positive side) set $h(\mathbf{x}) = 1$ if $\mathbf{w} \cdot \mathbf{x} + b > 0$.
- (Negative side) set $h(\mathbf{x}) = -1$ if $\mathbf{w} \cdot \mathbf{x} + b < 0$.

Given data with labels $y_i = \{\pm 1\}$, if there exists a hyperplane H so that, for all i, $\mathbf{x}_i$ has label 1 if and only if it is on the positive side of H, these data are called **linearly separable**.

Linearly separable

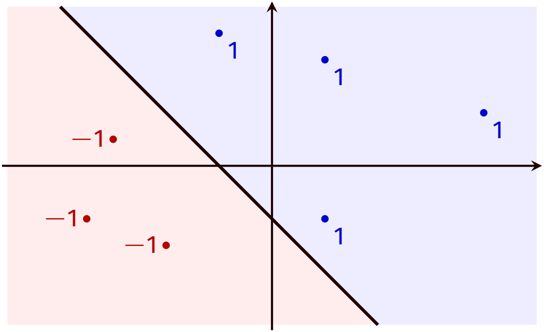


Figure: The hyperplane $H = \{(x_1, x_2) \in \mathbb{R}^2 : x_1 + x_2 + 1 = 0\}$, corresponding positive and negative regions, $\mathbf{w} = (1, 1)$, $b = 1$

Not linearly separable

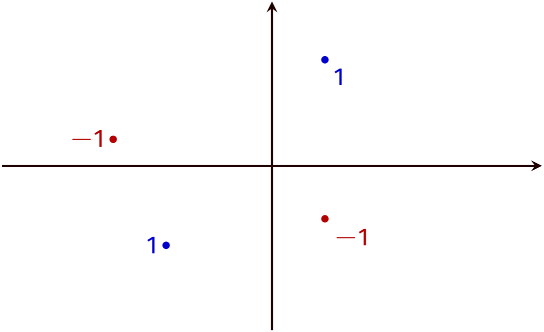


Figure: A data set in $\mathbb{R}^2$ that is not linearly separable.

Not linearly separable

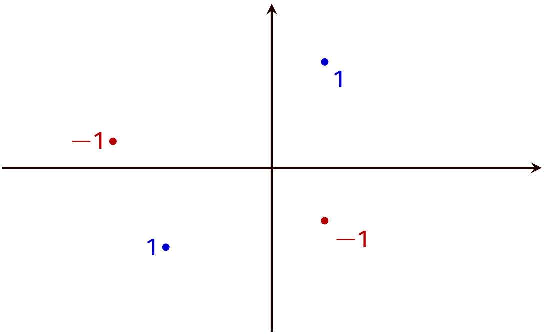


Figure: A data set in $\mathbb{R}^2$ that is not linearly separable.

- A criterion (checkable, in theory) that is equivalent to “not linearly separable”?

Outline

Classification tasks

Perceptron model

Perceptron algorithm

Setup for Perceptron algorithm

Labeled data: $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_P, y_P)$, with $\mathbf{x}_i \in \mathbb{R}^N$ and $y_i \in \{\pm 1\}$ for all i. Assuming labeled data is linearly separable, the Perceptron algorithm is a procedure that is guaranteed to find a hyperplane that separates the data.[4]

[4] Introduced in *The perceptron: A probabilistic model for information storage and organization in the brain*, F. Rosenblatt, Psychological Review **65** (1958), 386–407.

Setup for Perceptron algorithm

Labeled data: $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_P, y_P)$, with $\mathbf{x}_i \in \mathbb{R}^N$ and $y_i \in \{\pm 1\}$ for all i.
Assuming labeled data is linearly separable, the Perceptron algorithm is a procedure that is guaranteed to find a hyperplane that separates the data.[4]
To describe it: for each $\mathbf{x}_i$, use the notation $\tilde{\mathbf{w}}$ and $\tilde{\mathbf{x}}_i$ as before.

---

[4]Introduced in *The perceptron: A probabilistic model for information storage and organization in the brain*, F. Rosenblatt, Psychological Review **65** (1958), 386–407.

Setup for Perceptron algorithm

Labeled data: $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_P, y_P)$, with $\mathbf{x}_i \in \mathbb{R}^N$ and $y_i \in \{\pm 1\}$ for all i.
Assuming labeled data is linearly separable, the Perceptron algorithm is a procedure that is guaranteed to find a hyperplane that separates the data.[4]
To describe it: for each $\mathbf{x}_i$, use the notation $\tilde{\mathbf{w}}$ and $\tilde{\mathbf{x}}_i$ as before.
Note that $\tilde{\mathbf{w}} \cdot \tilde{\mathbf{x}}_i = \mathbf{w} \cdot \mathbf{x}_i + b$. For linearly separable data, our goal is to find $\tilde{\mathbf{w}} \in \mathbb{R}^{N+1}$ so that $\tilde{\mathbf{w}} \cdot \tilde{\mathbf{x}}_i$ and y_i have the same sign (both positive or both negative), for all $1 \leq i \leq P$.

- Equivalently, we need $y_i \, \tilde{\mathbf{w}} \cdot \tilde{\mathbf{x}}_i > 0$ for all $1 \leq i \leq P$.

[4] Introduced in *The perceptron: A probabilistic model for information storage and organization in the brain*, F. Rosenblatt, Psychological Review **65** (1958), 386–407.

Perceptron algorithm

Suppose the data is linearly separable. Also, make `x` be a $P \times N$ array of points, with i^{th} row equal to $\mathbf{x}_i$, and `y` an array of the labels. In the pseudocode below, use capitalization for the "tilde" notation: `W` is $\tilde{\mathbf{w}}$ and the i^{th} row of `X` is $\tilde{\mathbf{x}}_i$.
The Perceptron algorithm finds `W` iteratively as follows.[5]

[5] Recall, in pseudo-code block, left-facing arrow means *assign* to variable on left.

Perceptron algorithm

Suppose the data is linearly separable. Also, make x be a $P \times N$ array of points, with ith row equal to $\mathbf{x}_i$, and y an array of the labels. In the pseudocode below, use capitalization for the "tilde" notation: W is $\tilde{\mathbf{w}}$ and the i^{th} row of X is $\tilde{\mathbf{x}}_i$.

The Perceptron algorithm finds W iteratively as follows.[5]

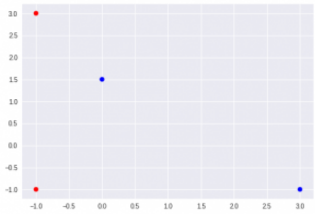
```
input: x, y  ## x is P by N, y is 1d array
X ← prepend 1 to each row of x
W ← (0,0,...,0)  ## Initial W
while (exists i with y[i]*dot(W, X[i]) ≤ 0){
    W ← W + y[i]*X[i]  # smallest such i
}
return W
```

[5]Recall, in pseudo-code block, left-facing arrow means *assign* to variable on left.

Example

A simple example in $\mathbb{R}^2$, with $n = 4$ points.

$$\text{x: } \begin{bmatrix} -1 & 3 \\ -1 & -1 \\ 3 & -1 \\ 0 & 1.5 \end{bmatrix} \qquad \text{y: } \begin{bmatrix} -1 \\ -1 \\ 1 \\ 1 \end{bmatrix}$$



Example, continued

A simple example in $\mathbb{R}^2$, with $n = 4$ points.

$$\text{x: } \begin{bmatrix} -1 & 3 \\ -1 & -1 \\ 3 & -1 \\ 0 & 1.5 \end{bmatrix} \qquad \text{y: } \begin{bmatrix} -1 \\ -1 \\ 1 \\ 1 \end{bmatrix}$$

Use $\tilde{\mathbf{w}}^{(t)}$ for value of $\tilde{\mathbf{w}}$ on step t. Start: $\tilde{\mathbf{w}}^{(1)} = (0, 0, 0)$.
Next step: $\tilde{\mathbf{w}}^{(2)} = \vec{0} + y_1 \tilde{\mathbf{x}}_1 = -1 * (1, -1, 3) = (-1, 1, -3)$.

Example, continued

A simple example in $\mathbb{R}^2$, with $n = 4$ points.

$$\text{x: } \begin{bmatrix} -1 & 3 \\ -1 & -1 \\ 3 & -1 \\ 0 & 1.5 \end{bmatrix} \qquad \text{y: } \begin{bmatrix} -1 \\ -1 \\ 1 \\ 1 \end{bmatrix}$$

Use $\tilde{\mathbf{w}}^{(t)}$ for value of $\tilde{\mathbf{w}}$ on step t. Start: $\tilde{\mathbf{w}}^{(1)} = (0, 0, 0)$.
Next step: $\tilde{\mathbf{w}}^{(2)} = \vec{0} + y_1\tilde{\mathbf{x}}_1 = -1 * (1, -1, 3) = (-1, 1, -3)$.
Next: since $y_1\tilde{\mathbf{w}}^{(2)} \cdot \tilde{\mathbf{x}}_1 > 0$, check
$y_2\tilde{\mathbf{w}}^{(2)} \cdot \tilde{\mathbf{x}}_2 = -1 * (-1 - 1 + 3) = -1$.

Example, continued

A simple example in $\mathbb{R}^2$, with $n = 4$ points.

$$\text{x:} \begin{bmatrix} -1 & 3 \\ -1 & -1 \\ 3 & -1 \\ 0 & 1.5 \end{bmatrix} \qquad \text{y:} \begin{bmatrix} -1 \\ -1 \\ 1 \\ 1 \end{bmatrix}$$

Use $\tilde{\mathbf{w}}^{(t)}$ for value of $\tilde{\mathbf{w}}$ on step t. Start: $\tilde{\mathbf{w}}^{(1)} = (0, 0, 0)$.
Next step: $\tilde{\mathbf{w}}^{(2)} = \vec{0} + y_1\tilde{\mathbf{x}}_1 = -1 * (1, -1, 3) = (-1, 1, -3)$.
Next: since $y_1\tilde{\mathbf{w}}^{(2)} \cdot \tilde{\mathbf{x}}_1 > 0$, check
$y_2\tilde{\mathbf{w}}^{(2)} \cdot \tilde{\mathbf{x}}_2 = -1 * (-1 - 1 + 3) = -1$. So,

$$\tilde{\mathbf{w}}^{(3)} = \tilde{\mathbf{w}}^{(2)} + y_2\tilde{\mathbf{x}}_2 = (-2, 2, -2).$$

Example, continued

A simple example in $\mathbb{R}^2$, with $n = 4$ points.

$$\text{x: } \begin{bmatrix} -1 & 3 \\ -1 & -1 \\ 3 & -1 \\ 0 & 1.5 \end{bmatrix} \qquad \text{y: } \begin{bmatrix} -1 \\ -1 \\ 1 \\ 1 \end{bmatrix}$$

Use $\tilde{\mathbf{w}}^{(t)}$ for value of $\tilde{\mathbf{w}}$ on step t. Start: $\tilde{\mathbf{w}}^{(1)} = (0, 0, 0)$.
Next step: $\tilde{\mathbf{w}}^{(2)} = \vec{0} + y_1\tilde{\mathbf{x}}_1 = -1 * (1, -1, 3) = (-1, 1, -3)$.
Next: since $y_1\tilde{\mathbf{w}}^{(2)} \cdot \tilde{\mathbf{x}}_1 > 0$, check
$y_2\tilde{\mathbf{w}}^{(2)} \cdot \tilde{\mathbf{x}}_2 = -1 * (-1 - 1 + 3) = -1$. So,

$$\tilde{\mathbf{w}}^{(3)} = \tilde{\mathbf{w}}^{(2)} + y_2\tilde{\mathbf{x}}_2 = (-2, 2, -2).$$

Continue in this way – on each step check dot products (in order) with $y_1\tilde{\mathbf{x}}_1, y_2\tilde{\mathbf{x}}_2, y_3\tilde{\mathbf{x}}_3, y_4\tilde{\mathbf{x}}_4$. Eventually you return the vector $\tilde{\mathbf{w}}^{(10)} = (1, 4, -0.5)$.

Example, continued

A simple example in $\mathbb{R}^2$, with $n = 4$ points.

$$\text{x:} \begin{bmatrix} -1 & 3 \\ -1 & -1 \\ 3 & -1 \\ 0 & 1.5 \end{bmatrix} \qquad \text{y:} \begin{bmatrix} -1 \\ -1 \\ 1 \\ 1 \end{bmatrix}$$

Use $\tilde{\mathbf{w}}^{(t)}$ for value of $\tilde{\mathbf{w}}$ on step t. Start: $\tilde{\mathbf{w}}^{(1)} = (0, 0, 0)$.
Next step: $\tilde{\mathbf{w}}^{(2)} = \vec{0} + y_1\tilde{\mathbf{x}}_1 = -1 * (1, -1, 3) = (-1, 1, -3)$.
Next: since $y_1\tilde{\mathbf{w}}^{(2)} \cdot \tilde{\mathbf{x}}_1 > 0$, check
$y_2\tilde{\mathbf{w}}^{(2)} \cdot \tilde{\mathbf{x}}_2 = -1 * (-1 - 1 + 3) = -1$. So,

$$\tilde{\mathbf{w}}^{(3)} = \tilde{\mathbf{w}}^{(2)} + y_2\tilde{\mathbf{x}}_2 = (-2, 2, -2).$$

Continue in this way – on each step check dot products (in order) with $y_1\tilde{\mathbf{x}}_1, y_2\tilde{\mathbf{x}}_2, y_3\tilde{\mathbf{x}}_3, y_4\tilde{\mathbf{x}}_4$. Eventually you return the vector $\tilde{\mathbf{w}}^{(10)} = (1, 4, -0.5)$.
i.e., $H = \{(x_1, x_2) \in \mathbb{R}^2 : \ 1 + 4x_1 - 0.5x_2 = 0\}$ separates the points.

Perceptron algorithm, stopping time

Under our assumptions for Perceptron algorithm, a guarantee on eventually stopping.

Theorem

Define $R = \max_i \|\tilde{\mathbf{x}}_i\|$ *and* $B = \min\{\|\mathbf{v}\| : \mathbf{v}$ *satisfies* , $y_i \mathbf{v} \cdot \tilde{\mathbf{x}}_i \geq 1, \forall i\}$. *Then, the Perceptron algorithm stops after at most* $(RB)^2$ *iterations and, when it stops with output* $\tilde{\mathbf{w}}$, *then* $y_i \tilde{\mathbf{w}} \cdot \tilde{\mathbf{x}}_i > 0$ *for all* $1 \leq i \leq P$.

Perceptron algorithm, stopping time

Under our assumptions for Perceptron algorithm, a guarantee on eventually stopping.

Theorem

Define $R = \max_i \|\tilde{\mathbf{x}}_i\|$ and $B = \min\{\|\mathbf{v}\| : \mathbf{v}$ satisfies , $y_i \mathbf{v} \cdot \tilde{\mathbf{x}}_i \geq 1, \forall i\}$. Then, the Perceptron algorithm stops after at most $(RB)^2$ iterations and, when it stops with output $\tilde{\mathbf{w}}$, then $y_i \tilde{\mathbf{w}} \cdot \tilde{\mathbf{x}}_i > 0$ for all $1 \leq i \leq P$.

Idea of proof: Write $\mathbf{v}^*$ for vector that realizes the minimum B. Also, $\tilde{\mathbf{w}}^{(t)}$ is the vector $\tilde{\mathbf{w}}$ on the t^{th} step, $\tilde{\mathbf{w}}^{(1)} = (0, 0, \ldots, 0)$.

Perceptron algorithm, stopping time

Under our assumptions for Perceptron algorithm, a guarantee on eventually stopping.

Theorem

Define $R = \max_i \|\tilde{\mathbf{x}}_i\|$ *and* $B = \min\{\|\mathbf{v}\| : \mathbf{v}$ *satisfies* , $y_i \mathbf{v} \cdot \tilde{\mathbf{x}}_i \geq 1, \forall i\}$. *Then, the Perceptron algorithm stops after at most* $(RB)^2$ *iterations and, when it stops with output* $\tilde{\mathbf{w}}$, *then* $y_i \tilde{\mathbf{w}} \cdot \tilde{\mathbf{x}}_i > 0$ *for all* $1 \leq i \leq P$.

Idea of proof: Write $\mathbf{v}^*$ for vector that realizes the minimum B. Also, $\tilde{\mathbf{w}}^{(t)}$ is the vector $\tilde{\mathbf{w}}$ on the t^{th} step, $\tilde{\mathbf{w}}^{(1)} = (0, 0, \ldots, 0)$.
Using how $\tilde{\mathbf{w}}^{(t+1)}$ is obtained from $\tilde{\mathbf{w}}^{(t)}$, can show that $\mathbf{v}^* \cdot \tilde{\mathbf{w}}^{(T+1)} \geq T$ after $T + 1$ iterations. Also, using the condition on $\tilde{\mathbf{w}}^{(T)}$ that necessitates an update, can show that $|\tilde{\mathbf{w}}^{(T+1)}| \leq R\sqrt{T}$. (For both statements, induction proves it.)

Perceptron algorithm, stopping time

Under our assumptions for Perceptron algorithm, a guarantee on eventually stopping.

Theorem

Define $R = \max_i \|\tilde{\mathbf{x}}_i\|$ *and* $B = \min\{\|\mathbf{v}\| : \mathbf{v}$ *satisfies* , $y_i \mathbf{v} \cdot \tilde{\mathbf{x}}_i \geq 1, \forall i\}$. *Then, the Perceptron algorithm stops after at most* $(RB)^2$ *iterations and, when it stops with output* $\tilde{\mathbf{w}}$, *then* $y_i \tilde{\mathbf{w}} \cdot \tilde{\mathbf{x}}_i > 0$ *for all* $1 \leq i \leq P$.

Idea of proof: Write $\mathbf{v}^*$ for vector that realizes the minimum B. Also, $\tilde{\mathbf{w}}^{(t)}$ is the vector $\tilde{\mathbf{w}}$ on the t^{th} step, $\tilde{\mathbf{w}}^{(1)} = (0, 0, \ldots, 0)$.
Using how $\tilde{\mathbf{w}}^{(t+1)}$ is obtained from $\tilde{\mathbf{w}}^{(t)}$, can show that $\mathbf{v}^* \cdot \tilde{\mathbf{w}}^{(T+1)} \geq T$ after $T+1$ iterations. Also, using the condition on $\tilde{\mathbf{w}}^{(T)}$ that necessitates an update, can show that $|\tilde{\mathbf{w}}^{(T+1)}| \leq R\sqrt{T}$. (For both statements, induction proves it.)
With those inequalities and the Cauchy-Schwarz inequality, $T \leq BR\sqrt{T}$, which we can rearrange to $T \leq (BR)^2$ (if an update was needed on step T).

Another example, the Iris data set

First discussed by R.A. Fisher in a 1936 paper, Iris data set commonly used in explanations. It contains 150 points in $\mathbb{R}^4$, each for an individual iris flower from one of 3 species: Iris setosa, Iris virginica, and Iris versicolor.

Figure: Images by G. Robertson, E. Hunt, Radomil ©CC BY-SA 3.0

Another example, the Iris data set

First discussed by R.A. Fisher in a 1936 paper, Iris data set commonly used in explanations. It contains 150 points in $\mathbb{R}^4$, each for an individual iris flower from one of 3 species: Iris setosa, Iris virginica, and Iris versicolor. The 4 coordinates are measurements of sepal length, sepal width, petal length, and petal width (in cm).

Figure: Images by G. Robertson, E. Hunt, Radomil ©CC BY-SA 3.0

Another example, the Iris data set

First discussed by R.A. Fisher in a 1936 paper, Iris data set commonly used in explanations. It contains 150 points in $\mathbb{R}^4$, each for an individual iris flower from one of 3 species: Iris setosa, Iris virginica, and Iris versicolor. The 4 coordinates are measurements of sepal length, sepal width, petal length, and petal width (in cm).

Iris setosa points are linearly separable from the other two.

Labels: *Iris setosa* $\leftarrow$ 1; *Other species* $\leftarrow$ -1.

Figure: Images by G. Robertson, E. Hunt, Radomil ©CC BY-SA 3.0

Another example, the Iris data set

First discussed by R.A. Fisher in a 1936 paper, Iris data set commonly used in explanations. It contains 150 points in $\mathbb{R}^4$, each for an individual iris flower from one of 3 species: Iris setosa, Iris virginica, and Iris versicolor. The 4 coordinates are measurements of sepal length, sepal width, petal length, and petal width (in cm).

Iris setosa points are linearly separable from the other two.

Labels: *Iris setosa* $\leftarrow$ 1; *Other species* $\leftarrow$ -1.

Begin by opening the notebook `'perceptron-iris-notebook.ipynb'` ...After completing the algorithm, should get final $\tilde{\mathbf{w}} = (b, \mathbf{w})$, where $\mathbf{w} = (1.3, 4.1, -5.2, -2.2)$ and $b = 1$.

Figure: Images by G. Robertson, E. Hunt, Radomil ©CC BY-SA 3.0