Logistic Regression

Chris Cornwell

Feb 27, 2025

Outline

Outline

Reconsidering the Half-space Model

Logistic model

Decision boundaries

For model h, made for classification task (with data points $\mathbf{x} \in \mathbb{R}^d$), write $C_y \subset \mathbb{R}^d$ for the set of points with label y, i.e.,

$$C_y = h^{-1}(y) = \{\mathbf{x} \in \mathbb{R}^d \mid h(\mathbf{x}) = y\}.$$

Decision boundaries

For model h, made for classification task (with data points $\mathbf{x} \in \mathbb{R}^d$), write $C_y \subset \mathbb{R}^d$ for the set of points with label y, i.e.,

$$C_y = h^{-1}(y) = \{\mathbf{x} \in \mathbb{R}^d \mid h(\mathbf{x}) = y\}.$$

Say $y \neq y'$ and a point is in the boundary of both C_y and $C_{y'}$. We say that point is on a **decision boundary** of the model. In a half-space model (last lecture), the hyperplane determined by $\mathbf{w}$ and b is the decision boundary.

Decision boundaries

For model h, made for classification task (with data points $\mathbf{x} \in \mathbb{R}^d$), write $C_y \subset \mathbb{R}^d$ for the set of points with label y, i.e.,

$$C_y = h^{-1}(y) = \{\mathbf{x} \in \mathbb{R}^d \mid h(\mathbf{x}) = y\}.$$

Say $y \neq y'$ and a point is in the boundary of both C_y and $C_{y'}$. We say that point is on a **decision boundary** of the model. In a half-space model (last lecture), the hyperplane determined by $\mathbf{w}$ and b is the decision boundary. The Perceptron algorithm might produce a model with many data points that are near the decision boundary.

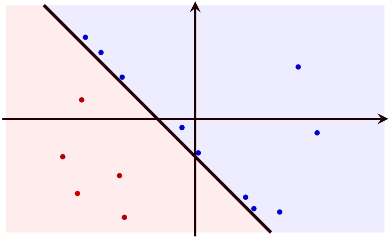


Figure: Many points near the decision boundary

But...data is messy

If many points close to the decision boundary, likely for newly observed data to appear on “wrong side” of decision boundary. Perceptron algorithm gives us nothing to correct for this.

But...data is messy

If many points close to the decision boundary, likely for newly observed data to appear on “wrong side” of decision boundary. Perceptron algorithm gives us nothing to correct for this.

If $\mathbf{x}$ is close to decision boundary, we might feel less confident in giving the label $h(\mathbf{x})$ that we do. And, if $\mathbf{x}$ is farther from the boundary, where only one label is seen nearby, more confidence is warranted.

But...data is messy

If many points close to the decision boundary, likely for newly observed data to appear on "wrong side" of decision boundary. Perceptron algorithm gives us nothing to correct for this.

If $\mathbf{x}$ is close to decision boundary, we might feel less confident in giving the label $h(\mathbf{x})$ that we do. And, if $\mathbf{x}$ is farther from the boundary, where only one label is seen nearby, more confidence is warranted.

Also...the immediate change of label across the boundary (a discontinuity in the model) ...perhaps not "natural"?

Outline

Reconsidering the Half-space Model

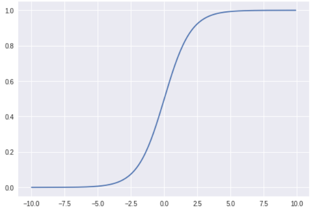
Logistic model

Incorporating a probability into half-space model

Instead of only capturing the sign of $\mathbf{w} \cdot \mathbf{x} + b$, compose it with the **logistic function**.

$$\sigma(z) = \frac{1}{1 + e^{-z}}.$$

- $0 < \sigma(z) < 1$ for all $z \in \mathbb{R}$;
- $\lim_{z \to \infty} \sigma(z) = 1$ and $\lim_{z \to -\infty} \sigma(z) = 0$;
- $\sigma(0) = 1/2$.



Logistic model, continued

"Logistic regression", used for binary classification, as follows.

Find hyperplane H, as determined by some $\mathbf{w}$ and b, that fits labeled data well; given new $\mathbf{x} \in \mathbb{R}^d$, find $z = \mathbf{w} \cdot \mathbf{x} + b$, then compute $\sigma(z)$.

- (Logistic regression) $\sigma(z)$, the *probability* that the label is $+1$.

[1]Since -1 is only other label, high probability of having label -1.

Logistic model, continued

"Logistic regression", used for binary classification, as follows.

Find hyperplane H, as determined by some $\mathbf{w}$ and b, that fits labeled data well; given new $\mathbf{x} \in \mathbb{R}^d$, find $z = \mathbf{w} \cdot \mathbf{x} + b$, then compute $\sigma(z)$.

- (Logistic regression) $\sigma(z)$, the *probability* that the label is $+1$.

1. If $\mathbf{w} \cdot \mathbf{x} + b > 0$ is very large ($\mathbf{x}$ far away from H and on positive side), then $\sigma(z)$ is very close to 1.
2. If $\mathbf{w} \cdot \mathbf{x} + b < 0$ has abs. value very large ($\mathbf{x}$ far away from H on negative side), then $\sigma(z)$ very close to 0.[1]
3. If $\mathbf{x}$ is contained in H itself, $z = 0$ and $\sigma(z) = 0.5$.

[1] Since -1 is only other label, high probability of having label -1.

Logistic model, continued

"Logistic regression", used for binary classification, as follows.
Find hyperplane H, as determined by some $\mathbf{w}$ and b, that fits labeled data well; given new $\mathbf{x} \in \mathbb{R}^d$, find $z = \mathbf{w} \cdot \mathbf{x} + b$, then compute $\sigma(z)$.

- (Logistic regression) $\sigma(z)$, the *probability* that the label is $+1$.

1. If $\mathbf{w} \cdot \mathbf{x} + b > 0$ is very large ($\mathbf{x}$ far away from H and on positive side), then $\sigma(z)$ is very close to 1.
2. If $\mathbf{w} \cdot \mathbf{x} + b < 0$ has abs. value very large ($\mathbf{x}$ far away from H on negative side), then $\sigma(z)$ very close to 0.[1]
3. If $\mathbf{x}$ is contained in H itself, $z = 0$ and $\sigma(z) = 0.5$.

Binary classifier model from logistic regression: $h(\mathbf{x}) = 1$ if $\sigma(\mathbf{w} \cdot \mathbf{x} + b) \geq 0.5$, and $h(\mathbf{x}) = -1$ otherwise.

- *Remember that probability (certainty) in the prediction.*

---

[1] Since -1 is only other label, high probability of having label -1.

Logistic model, continued

"Logistic regression", used for binary classification, as follows.

Find hyperplane H, as determined by some $\mathbf{w}$ and b, that fits labeled data well; given new $\mathbf{x} \in \mathbb{R}^d$, find $z = \mathbf{w} \cdot \mathbf{x} + b$, then compute $\sigma(z)$.

- (Logistic regression) $\sigma(z)$, the *probability* that the label is $+1$.

Binary classifier model from logistic regression: $h(\mathbf{x}) = 1$ if $\sigma(\mathbf{w} \cdot \mathbf{x} + b) \geq 0.5$, and $h(\mathbf{x}) = -1$ otherwise.

Logistic model, continued

"Logistic regression", used for binary classification, as follows.
Find hyperplane H, as determined by some $\mathbf{w}$ and b, that fits labeled data well; given new $\mathbf{x} \in \mathbb{R}^d$, find $z = \mathbf{w} \cdot \mathbf{x} + b$, then compute $\sigma(z)$.

- (Logistic regression) $\sigma(z)$, the *probability* that the label is $+1$.

Binary classifier model from logistic regression: $h(\mathbf{x}) = 1$ if $\sigma(\mathbf{w} \cdot \mathbf{x} + b) \geq 0.5$, and $h(\mathbf{x}) = -1$ otherwise.
Some rationale for the use of the logistic function $\frac{1}{1+e^{-z}}$ in this process, comes from an inverse direction.

- If wanting to use *Maximum Likelihood Estimation* to get a binary model, and making some typical simplifying assumptions on conditional probability (P), given model parameters, of observing some $\mathbf{x_i} \rightsquigarrow \log\left(\frac{P}{1-P}\right)$ being linear.

How to find $\mathbf{w}$ and b

Given $\{\pm 1\}$ labeled data, how do we go about finding $\mathbf{w}$ and b to use in the logistic (regression) model?

- Even if data is linearly separable, Perceptron algorithm does not try to make hyperplane be positioned "away from" data (Disadvantage).
- If data is not linearly separable, what should be done?

Future lectures: Will discuss using optimization (calculus-based) to find best parameters $w_1, w_2, \ldots, w_d, b$; process called Gradient Descent.

- Relevant: relationship between gradient of a function and its directional derivative.

More messy versus less messy data

Could introduce additional parameter (more flexibility).

For $k > 0$, define

$$\sigma_k(z) = \frac{1}{1 + e^{-kz}}.$$

$0 < k < 1$: values of $\sigma_k(z)$ transition from 0 to 1 more slowly.

$k > 1$: values of $\sigma_k(z)$ transition from 0 to 1 more quickly. (think about derivative)

Hence, if know data has more noise, might use $0 < k < 1$ to decrease measure of confidence in prediction. In contrast, very "clean" data, interpretation of the model might benefit from $k > 1$.

(**Left:** graph with $k = 5$; **Right:** applied to points in $\mathbb{R}^2$)