

COMPSCI 689

Lecture 4: Logistic Regression

Brendan O'Connor

College of Information and Computer Sciences
University of Massachusetts Amherst

Slides by Benjamin M. Marlin (marlin@cs.umass.edu).

Outline

1 Review

2 The Linear Classifier

3 Logistic Regression

Empirical Risk Minimization

Let Θ be the parameter space for a set \mathcal{F} of prediction functions f_θ mapping from \mathcal{X} to \mathcal{Y} . The principle of Empirical Risk Minimization (ERM) states that we should select the parameters $\theta \in \Theta$ that *minimize the average of the prediction loss* $L(\mathbf{y}_n, f_\theta(\mathbf{x}_n))$ computed over the data set \mathcal{D} , also known as the empirical risk $R(f_\theta, \mathcal{D})$:

$$\hat{\theta} = \arg \min_{\theta \in \Theta} R(f_\theta, \mathcal{D})$$

$$R(f_\theta, \mathcal{D}) = \frac{1}{N} \sum_{n=1}^N L(\mathbf{y}_n, f_\theta(\mathbf{x}_n))$$

Binary Classification

In binary classification, the output y can be one of two possible values. The output y is referred to as the “class” or “label” of \mathbf{x} . The inputs \mathbf{x} are also referred to as “features.” The goal is to predict the class given the features.

- Input Space: \mathcal{X}
- Output Space: $\mathcal{Y} = \{0, 1\}$ or $\{-1, 1\}$
- Input: $\mathbf{x} \in \mathcal{X}$
- Output: $\mathbf{y} \in \mathcal{Y}$
- Prediction Function: $f: \mathcal{X} \rightarrow \mathcal{Y}$

Binary Classification

In binary classification, the output y can be one of two possible values. The output y is referred to as the “class” or “label” of \mathbf{x} . The inputs \mathbf{x} are also referred to as “features.” The goal is to predict the class given the features.

- Input Space: \mathcal{X}
- Output Space: $\mathcal{Y} = \{0, 1\}$ or $\{-1, 1\}$
- Input: $\mathbf{x} \in \mathcal{X}$
- Output: $\mathbf{y} \in \mathcal{Y}$
- Prediction Function: $f: \mathcal{X} \rightarrow \mathcal{Y}$

Question: What are some examples of binary classification tasks?

Outline

1 Review

2 The Linear Classifier

3 Logistic Regression

The Linear Classifier

- Suppose $y \in \{-1, 1\}$ and $\mathbf{x} \in \mathbb{R}^D$.

The Linear Classifier

- Suppose $y \in \{-1, 1\}$ and $\mathbf{x} \in \mathbb{R}^D$.
- Let $\theta = [\mathbf{w}; b]$ where $\mathbf{w} \in \mathbb{R}^D, b \in \mathbb{R}$.

The Linear Classifier

- Suppose $y \in \{-1, 1\}$ and $\mathbf{x} \in \mathbb{R}^D$.
- Let $\theta = [\mathbf{w}; b]$ where $\mathbf{w} \in \mathbb{R}^D$, $b \in \mathbb{R}$.
- \mathbf{w} are referred to as the weights, b is the bias.

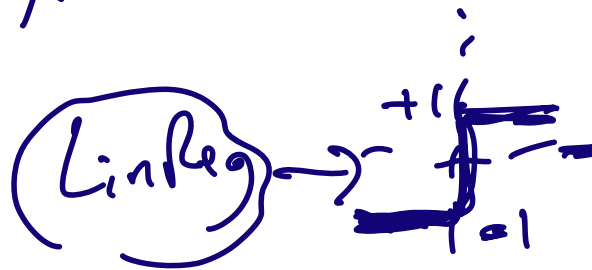
The Linear Classifier

- Suppose $y \in \{-1, 1\}$ and $\mathbf{x} \in \mathbb{R}^D$.
- Let $\theta = [\mathbf{w}; b]$ where $\mathbf{w} \in \mathbb{R}^D$, $b \in \mathbb{R}$.
- \mathbf{w} are referred to as the weights, b is the bias.

- A linear classifier uses a prediction function of the form:

$$f_{\theta}(\mathbf{x}) = \text{sign}(\mathbf{x}\mathbf{w} + b)$$

$$\text{sign}(z) = \begin{cases} +1 & \text{if } z \geq 0 \\ -1 & \text{if } z < 0 \end{cases}$$



The Linear Classifier

- Suppose $y \in \{-1, 1\}$ and $\mathbf{x} \in \mathbb{R}^D$.
- Let $\theta = [\mathbf{w}; b]$ where $\mathbf{w} \in \mathbb{R}^D$, $b \in \mathbb{R}$.
- \mathbf{w} are referred to as the weights, b is the bias.
- A linear classifier uses a prediction function of the form:

$$f_{\theta}(\mathbf{x}) = \text{sign}(\mathbf{x}\mathbf{w} + b)$$

- The function $g_{\theta}(\mathbf{x}) = \mathbf{x}\mathbf{w} + b$ is referred to as the discriminant function.

Linear Classifier Geometry

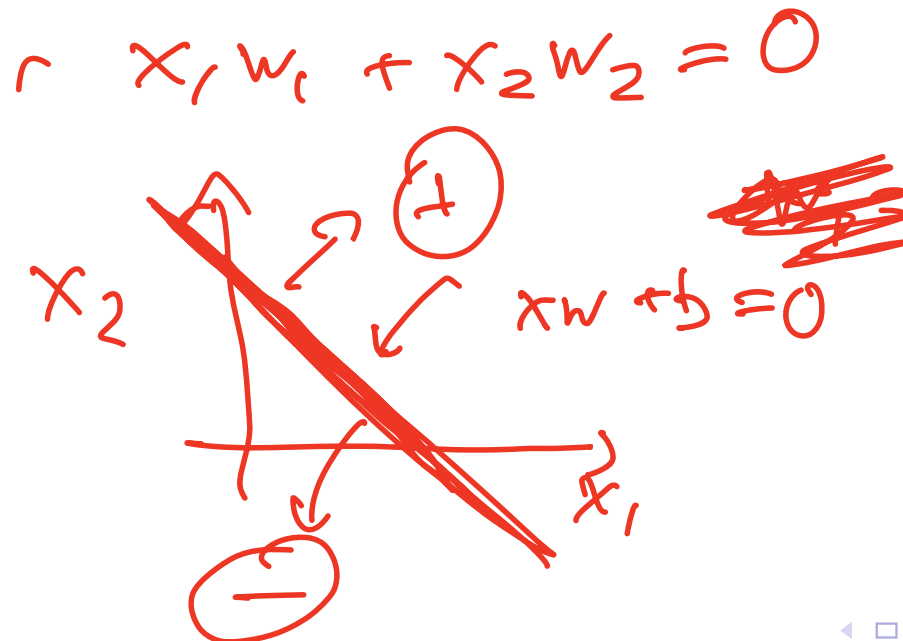
- Points satisfying $g_{\theta}(\mathbf{x}) = \mathbf{xw} + b > 0$ are predicted to be in the positive class.

Linear Classifier Geometry

- Points satisfying $g_{\theta}(\mathbf{x}) = \mathbf{xw} + b > 0$ are predicted to be in the positive class.
- Points satisfying $g_{\theta}(\mathbf{x}) = \mathbf{xw} + b < 0$ are predicted to be in the negative class.

Linear Classifier Geometry

- Points satisfying $g_{\theta}(\mathbf{x}) = \mathbf{xw} + b > 0$ are predicted to be in the positive class.
- Points satisfying $g_{\theta}(\mathbf{x}) = \mathbf{xw} + b < 0$ are predicted to be in the negative class.
- Points satisfying $g_{\theta}(\mathbf{x}) = \mathbf{xw} + b = 0$ are on the *decision boundary* between classes. $\mathbf{xw} = -b$



Linear Classifier Geometry

- Points satisfying $g_{\theta}(\mathbf{x}) = \mathbf{x}\mathbf{w} + b > 0$ are predicted to be in the positive class.
- Points satisfying $g_{\theta}(\mathbf{x}) = \mathbf{x}\mathbf{w} + b < 0$ are predicted to be in the negative class.
- Points satisfying $g_{\theta}(\mathbf{x}) = \mathbf{x}\mathbf{w} + b = 0$ are on the *decision boundary* between classes.
- The set of points such that $g_{\theta}(\mathbf{x}) = \mathbf{x}\mathbf{w} + b = 0$ forms a hyperplane in \mathbb{R}^D (a straight line in 2D, a plane in 3D, etc...).

The Linear Classifier and Bias Absorption

- We can again use bias absorption to incorporate b into \mathbf{w} :

The Linear Classifier and Bias Absorption

- We can again use bias absorption to incorporate b into \mathbf{w} :
- If we append a 1 to the input vector forming $\tilde{\mathbf{x}} = [\mathbf{x}, 1]$, then we have: $\mathbf{xw} + b = \tilde{\mathbf{x}}\theta$.

The Linear Classifier and Bias Absorption

- We can again use bias absorption to incorporate b into \mathbf{w} :
- If we append a 1 to the input vector forming $\tilde{\mathbf{x}} = [\mathbf{x}, 1]$, then we have: $\mathbf{x}\mathbf{w} + b = \tilde{\mathbf{x}}\theta$.
- As with linear regression, we'll assume bias absorption to simplify the description of the model.

Classification Loss

- **Question:** What loss function should we use?

Classification Loss

- **Question:** What loss function should we use?
- The most natural prediction loss to use in the classification setting is classification error or zero-one loss.

Classification Loss

- **Question:** What loss function should we use?
- The most natural prediction loss to use in the classification setting is classification error or zero-one loss.
- Using the prediction function, we have $L(y, f_{\theta}(\mathbf{x})) = [y \neq f_{\theta}(\mathbf{x})]$.

*Indicate
fun*

Classification Loss

- **Question:** What loss function should we use?
- The most natural prediction loss to use in the classification setting is classification error or zero-one loss.
- Using the prediction function, we have $L(y, f_{\theta}(\mathbf{x})) = [y \neq f_{\theta}(\mathbf{x})]$.
- If we predict the wrong class, we incur a loss of 1, otherwise we incur a loss of 0.

Classification Loss

- **Question:** What loss function should we use?
- The most natural prediction loss to use in the classification setting is classification error or zero-one loss.
- Using the prediction function, we have $L(y, f_{\theta}(\mathbf{x})) = [y \neq f_{\theta}(\mathbf{x})]$.
- If we predict the wrong class, we incur a loss of 1, otherwise we incur a loss of 0.
- We can also use the discriminant function and write $L(y, f_{\theta}(\mathbf{x})) = [y \cdot g_{\theta}(\mathbf{x}) < 0]$.

no loss when: $\begin{matrix} \downarrow & \downarrow \\ + & + \end{matrix}$ ☺

$\begin{matrix} - & - \\ + & - \\ - & + \end{matrix}$ ☹
2/3 = 0.66

$y = +1$ want $g(x) > 0$

Classification Loss

- **Question:** What loss function should we use?
- The most natural prediction loss to use in the classification setting is classification error or zero-one loss.
- Using the prediction function, we have $L(y, f_{\theta}(\mathbf{x})) = [y \neq f_{\theta}(\mathbf{x})]$.
- If we predict the wrong class, we incur a loss of 1, otherwise we incur a loss of 0.
- We can also use the discriminant function and write $L(y, f_{\theta}(\mathbf{x})) = [y \cdot g_{\theta}(\mathbf{x}) < 0]$.
- If the signs of y and $g_{\theta}(\mathbf{x})$ disagree, then we predicted that \mathbf{x} is on the opposite side of the decision boundary from y and we incur a loss of 1, otherwise we incur a loss of 0.

ERM for the Linear Classifier

Given the choice of classification error as the prediction loss $L(y, y') = [y \neq y']$ and the space of prediction functions $\mathcal{F} = \{f_\theta(\mathbf{x}) = \text{sign}(\mathbf{x}\theta) | \theta \in \Theta\}$, we can apply ERM to define the optimal prediction function parameters:

$$\hat{\theta} = \arg \min_{\theta \in \Theta} R(\theta, \mathcal{D})$$

ERM for the Linear Classifier

Given the choice of classification error as the prediction loss $L(y, y') = [y \neq y']$ and the space of prediction functions $\mathcal{F} = \{f_\theta(\mathbf{x}) = \text{sign}(\mathbf{x}\theta) | \theta \in \Theta\}$, we can apply ERM to define the optimal prediction function parameters:

$$\hat{\theta} = \arg \min_{\theta \in \Theta} R(\theta, \mathcal{D})$$

$$R(\theta, \mathcal{D}) = \frac{1}{N} \sum_{n=1}^N [y_n \neq \text{sign}(\mathbf{x}_n \theta)]$$

ERM for the Linear Classifier

Given the choice of classification error as the prediction loss $L(y, y') = [y \neq y']$ and the space of prediction functions $\mathcal{F} = \{f_\theta(\mathbf{x}) = \text{sign}(\mathbf{x}\theta) | \theta \in \Theta\}$, we can apply ERM to define the optimal prediction function parameters:

$$\hat{\theta} = \arg \min_{\theta \in \Theta} R(\theta, \mathcal{D})$$

$$R(\theta, \mathcal{D}) = \frac{1}{N} \sum_{n=1}^N [y_n \neq \text{sign}(\mathbf{x}_n \theta)]$$



Question: How do we actually find the model parameters θ that minimize the empirical risk defined above?

Outline

1 Review

2 The Linear Classifier

3 Logistic Regression

Alternative Loss Functions

- The problem with using classification error as the loss is that the resulting ERM objective function is not differentiable, so it is not amenable to optimization methods based on stationary point analysis.

Alternative Loss Functions

- The problem with using classification error as the loss is that the resulting ERM objective function is not differentiable, so it is not amenable to optimization methods based on stationary point analysis.
- One approach to addressing this problem is to minimize a differentiable upper-bound on the classification error instead of the classification error itself.

Alternative Loss Functions

- The problem with using classification error as the loss is that the resulting ERM objective function is not differentiable, so it is not amenable to optimization methods based on stationary point analysis.
- One approach to addressing this problem is to minimize a differentiable upper-bound on the classification error instead of the classification error itself.
- The advantage of minimizing an upper bound is that if we can make the upper bound small, we know that the actual loss that we care about is no larger.

Logistic Loss

- The most commonly used differentiable upper bound on classification error is the logistic loss function:

$$L(y, g_{\theta}(\mathbf{x})) = \frac{1}{\log(2)} \log(1 + \exp(-y \cdot g_{\theta}(\mathbf{x})))$$

$$L_{LO} = 1 \{ y \cdot g(\mathbf{x}) < 0 \}$$

Logistic Loss

- The most commonly used differentiable upper bound on classification error is the logistic loss function:

$$L(y, g_{\theta}(\mathbf{x})) = \frac{1}{\log(2)} \log(1 + \exp(-y \cdot g_{\theta}(\mathbf{x})))$$

- If the signs of y and $g_{\theta}(\mathbf{x})$ agree, then as $g_{\theta}(\mathbf{x})$ increases in magnitude, the loss decays smoothly to 0.

Logistic Loss

- The most commonly used differentiable upper bound on classification error is the logistic loss function:

$$L(y, g_{\theta}(\mathbf{x})) = \frac{1}{\log(2)} \log(1 + \exp(-y \cdot g_{\theta}(\mathbf{x})))$$

- If the signs of y and $g_{\theta}(\mathbf{x})$ agree, then as $g_{\theta}(\mathbf{x})$ increases in magnitude, the loss decays smoothly to 0.
- If the signs of y and $g_{\theta}(\mathbf{x})$ disagree, then as $g_{\theta}(\mathbf{x})$ increases in magnitude, the loss smoothly converges to a linearly increasing function.

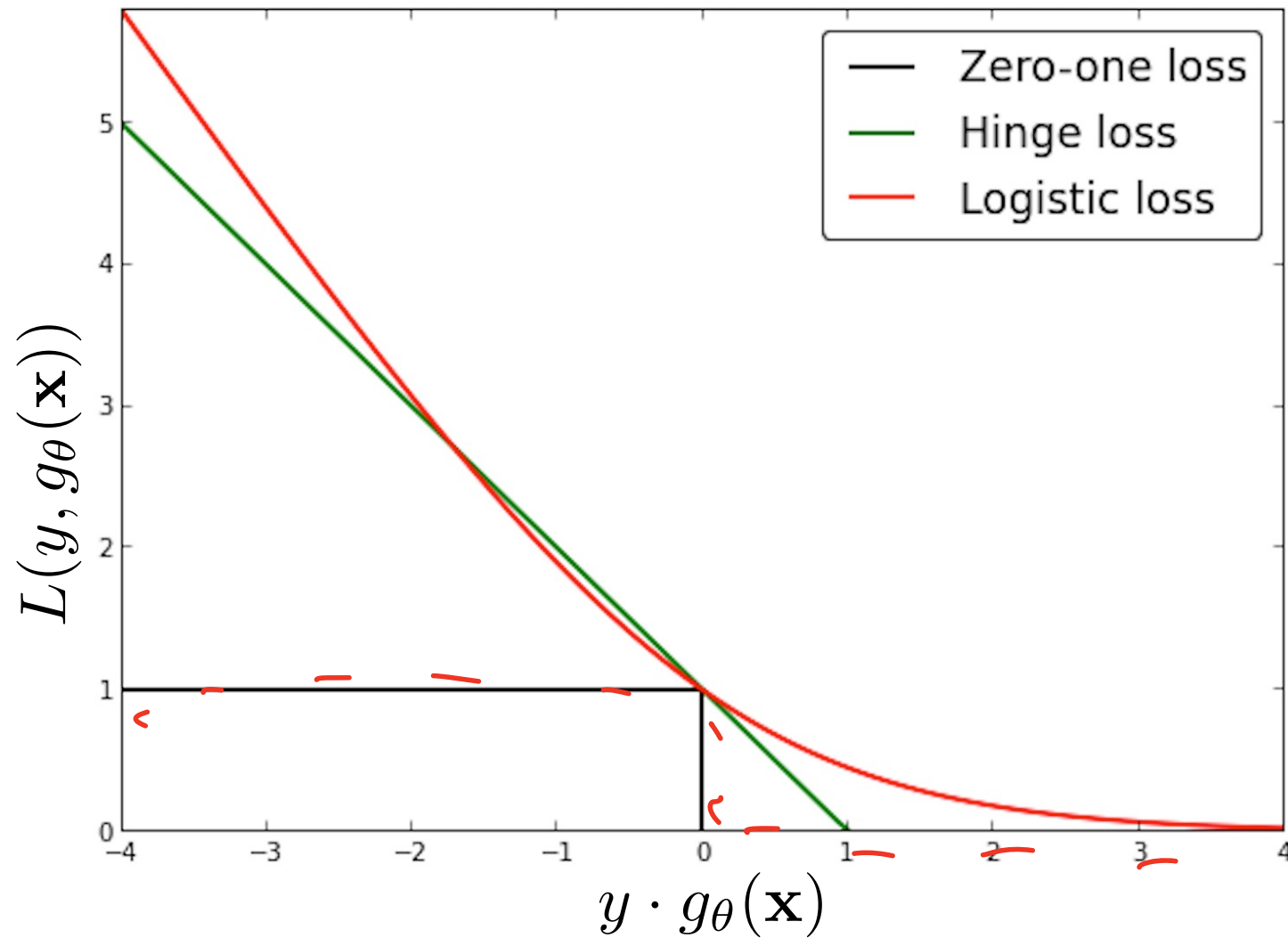
Logistic Loss

- The most commonly used differentiable upper bound on classification error is the logistic loss function:

$$L(y, g_{\theta}(\mathbf{x})) = \frac{1}{\log(2)} \log(1 + \exp(-y \cdot g_{\theta}(\mathbf{x})))$$

- If the signs of y and $g_{\theta}(\mathbf{x})$ agree, then as $g_{\theta}(\mathbf{x})$ increases in magnitude, the loss decays smoothly to 0.
- If the signs of y and $g_{\theta}(\mathbf{x})$ disagree, then as $g_{\theta}(\mathbf{x})$ increases in magnitude, the loss smoothly converges to a linearly increasing function.
- To provide an upper bound, the $1 / \log(2)$ term is required, but since it doesn't affect the location of the optimal parameters, it is most often dropped during learning.

Classification Losses



Logistic loss: Probabilistic view

Logistic sigmoid function $\sigma(z) : \mathbb{R} \rightarrow [0, 1]$:

$$\sigma(z) = \frac{e^z}{1 + e^z} = \frac{1}{1 + e^{-z}} = \text{graph of sigmoid function}$$

$$p(y = 1 \mid \mathbf{x}) = \sigma(g(\mathbf{x})) \quad \begin{array}{l} g(\mathbf{x}) = 0 \text{ dec boundary} \\ \Leftrightarrow p(y=1/\mathbf{x}) = .5 \end{array}$$

Logistic loss is the negative log-likelihood of this model

$$\begin{aligned} \log p(Y/X) &= \sum_n \log p(y_n/x_n) \\ &= \sum_n \log \begin{cases} \sigma(g(x_n)) & \text{if } y = +1 \\ 1 - \sigma(g(x_n)) & \text{if } y = -1 \end{cases} \end{aligned}$$

$$\begin{aligned} \log \sigma(g(x)) &= -\log(1 + e^{-g(x)}) \end{aligned}$$

$$\Rightarrow = -\mathcal{L}_{\text{log loss}}(Y, g)$$

ERM for the Linear Classifier: Take 2

Given the choice of the logistic loss as an upper bound on the classification loss and the space of discriminant functions $\mathcal{G} = \{g_{\theta}(\mathbf{x}) = \mathbf{x}\theta \mid \theta \in \Theta\}$, we can now apply ERM to define the optimal prediction function parameters. The resulting method is referred to as *logistic regression*.

$$\hat{\theta} = \arg \min_{\theta \in \Theta} R(\theta, \mathcal{D})$$

ERM for the Linear Classifier: Take 2

Given the choice of the logistic loss as an upper bound on the classification loss and the space of discriminant functions $\mathcal{G} = \{g_{\theta}(\mathbf{x}) = \mathbf{x}\theta \mid \theta \in \Theta\}$, we can now apply ERM to define the optimal prediction function parameters. The resulting method is referred to as *logistic regression*.

$$\hat{\theta} = \arg \min_{\theta \in \Theta} R(\theta, \mathcal{D})$$

$$R(\theta, \mathcal{D}) = \frac{1}{N} \sum_{n=1}^N \log(1 + \exp(-y_n \cdot g_{\theta}(\mathbf{x}_n)))$$

ERM for the Linear Classifier: Take 2

Given the choice of the logistic loss as an upper bound on the classification loss and the space of discriminant functions $\mathcal{G} = \{g_{\theta}(\mathbf{x}) = \mathbf{x}\theta \mid \theta \in \Theta\}$, we can now apply ERM to define the optimal prediction function parameters. The resulting method is referred to as *logistic regression*.

$$\hat{\theta} = \arg \min_{\theta \in \Theta} R(\theta, \mathcal{D})$$

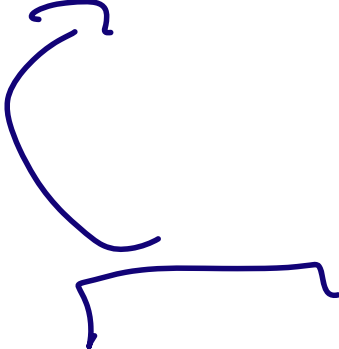
$$R(\theta, \mathcal{D}) = \frac{1}{N} \sum_{n=1}^N \log(1 + \exp(-y_n \cdot g_{\theta}(\mathbf{x}_n)))$$

Question: How do we actually find the model parameters θ that minimize the empirical risk defined above?

Step 1: Derive Gradient

$$\begin{aligned}\nabla R(\theta, \mathcal{D}) &= \nabla \frac{1}{N} \sum_{n=1}^N \log(1 + \exp(-y_n \cdot g_{\theta}(\mathbf{x}_n))) \\&= \frac{1}{N} \sum_{n=1}^N \nabla \log(1 + \exp(-y_n \cdot g_{\theta}(\mathbf{x}_n))) \\&= \frac{1}{N} \sum_{n=1}^N \frac{1}{1 + \exp(-y_n \cdot g_{\theta}(\mathbf{x}_n))} \exp(-y_n \cdot g_{\theta}(\mathbf{x}_n)) (-y_n \cdot \nabla g_{\theta}(\mathbf{x}_n)) \\&= \frac{1}{N} \sum_{n=1}^N \frac{1}{1 + \exp(-y_n \cdot g_{\theta}(\mathbf{x}_n))} \exp(-y_n \cdot g_{\theta}(\mathbf{x}_n)) (-y_n \cdot \mathbf{x}_n^T) \\&= -\frac{1}{N} \sum_{n=1}^N y_n \frac{\exp(-y_n \cdot g_{\theta}(\mathbf{x}_n))}{1 + \exp(-y_n \cdot g_{\theta}(\mathbf{x}_n))} \mathbf{x}_n^T\end{aligned}$$

$\nabla_{\theta}(x^{\theta}) = x$



Step 1: Derive Gradient

$$\begin{aligned} &= -\frac{1}{N} \sum_{n=1}^N y_n \frac{\exp(-y_n \cdot g_{\theta}(\mathbf{x}_n))}{1 + \exp(-y_n \cdot g_{\theta}(\mathbf{x}_n))} \mathbf{x}_n^T \\ &= -\frac{1}{N} \sum_{n=1}^N y_n \left(1 - \frac{1}{1 + \exp(-y_n \cdot g_{\theta}(\mathbf{x}_n))} \right) \mathbf{x}_n^T \\ &= -\frac{1}{N} \sum_{n=1}^N y_n (1 - \sigma(y_n \cdot g_{\theta}(\mathbf{x}_n))) \mathbf{x}_n^T \\ &= \frac{1}{N} \sum_{n=1}^N y_n (\sigma(y_n \cdot g_{\theta}(\mathbf{x}_n)) - 1) \mathbf{x}_n^T \end{aligned}$$

Step 2: Solve Gradient Equation

$$\nabla R(\theta, \mathcal{D}) = \frac{1}{N} \sum_{n=1}^N y_n (\sigma(y_n \cdot g_{\theta}(\mathbf{x}_n)) - 1) \mathbf{x}_n^T = 0$$

Logistic Regression by Steepest Descent

- 1 Choose $\theta_0, \alpha > 0$,
- 2 On each iteration k :
 - 1 Compute $\mathbf{d}_k = -\nabla R(\theta_k, \mathcal{D})$
 - 2 $\theta_{k+1} \leftarrow \theta_k + \alpha \mathbf{d}_k$

