**CS 4850/6850: Introduction to Machine Learning** **Fall 2018**

# Topic 2: Logistic Regression

Instructor: Daniel L. Pimentel-Alarcón

## 2.1 Introduction

Arguably the simplest task that we can teach a computer is to distinguish between two classes. For example:

1. Does this image contain a dog or a cat?
2. Is this person healthy or diabetic?
3. Would this individual survive a disaster?

Logistic regression is one of the most elemental yet powerful techniques for this purpose. The main idea is to compute the *likelihood* that a sample (e.g., a person) belongs to each class, based on its information and the information of previous (training) samples, and then choose the most likely class.

## 2.2 Setup

Suppose you want to determine whether your girlfriend/boyfriend is cheating on you, based on certain information (features) about her/him, like age, gender, height, weight, etc. Let $\mathbf{x}_\mathrm{i} \in \mathbb{R}^\mathrm{d}$ denote the vector containing this information, which looks like this:

$$\mathbf{x}_\mathrm{i} = \begin{bmatrix} age \\ gender \\ height \\ weight \\ \vdots \end{bmatrix}.$$

Here d denotes the number of features. Similarly, let $y$ be the random variable indicating whether s/he is not cheating on you: $y = 1$ means s/he is, $y = 0$ means s/he isn't. Hence we can rephrase our goal as determining whether $y = 0$ or $y = 1$ based on $\mathbf{x}$. Mathematically, we want to find a function $f$ such that

$$y \;=\; f(\mathbf{x}).$$

Perhaps the most natural way to achieve this is to let $f$ be of the form:

$$f(\mathbf{x}) \;=\; \begin{cases} 0 & \text{if } \mathbb{P}(y=0|\mathbf{x}) > \mathbb{P}(y=1|\mathbf{x}) \\ 1 & \text{otherwise.} \end{cases} \tag{2.1}$$

In words, (2.1) is simply saying: decide $y = 0$ if the probability of $y$ being 0 (based on $\mathbf{x}$) is larger than the probability of $y$ being 1, and decide $y = 1$ otherwise. We can rewrite this as follows:

$$\mathbb{P}(y = 1|\mathbf{x}) \underset{f(\mathbf{x})=0}{\overset{f(\mathbf{x})=1}{\gtrless}} \mathbb{P}(y = 0|\mathbf{x}),$$

or equivalently, as:

$$\frac{\mathbb{P}(y = 1|\mathbf{x})}{\mathbb{P}(y = 0|\mathbf{x})} \underset{f(\mathbf{x})=0}{\overset{f(\mathbf{x})=1}{\gtrless}} 1.$$

The term on the left is often known as the *odds*. If we know the odds, we know whether $\mathbb{P}(y = 1|\mathbf{x})$ or $\mathbb{P}(y = 0|\mathbf{x})$ is more likely, and we can decide accordingly. Hence, our goal is to determining what are the odds, based on $\mathbf{x}$. Arguably, the simplest, most natural approach is to model the odds as a linear combination of the entries in $\mathbf{x}$, i.e.,

$$\frac{\mathbb{P}(y = 1|\mathbf{x})}{\mathbb{P}(y = 0|\mathbf{x})} = \boldsymbol{\beta}^\mathsf{T}\mathbf{x}, \tag{2.2}$$

where $\boldsymbol{\beta} \in \mathbb{R}^{\mathrm{d}}$ contains the *coefficients* of the linear combination of $\mathbf{x}$. Notice that $\boldsymbol{\beta}^\mathsf{T}$ is just the compact (grown up) way to write $\beta_1\mathrm{x}_1 + \beta_2\mathrm{x}_2 + \cdots + \beta_{\mathrm{d}}\mathrm{x}_{\mathrm{d}}$. The problem with (2.2) is that $\frac{\mathbb{P}(y=1|\mathbf{x})}{\mathbb{P}(y=0|\mathbf{x})} \geq 0$, while $\boldsymbol{\beta}^\mathsf{T}\mathbf{x} \in \mathbb{R}$. To avoid this discrepancy, rather than (2.2), logistic regression simply applies a log function on the odds, to obtain:

$$\log\left(\frac{\mathbb{P}(y = 1|\mathbf{x})}{\mathbb{P}(y = 0|\mathbf{x})}\right) = \boldsymbol{\beta}^\mathsf{T}\mathbf{x}. \tag{2.3}$$

The term on the left is often known as *log-odds*. It is from this idea that logistic regression obtains its name. Notice that in (2.3), both the log-odds and $\boldsymbol{\beta}^\mathsf{T}\mathbf{x}$ are real numbers, so there is no longer any discrepancy. Letting $\mathrm{p} := \mathbb{P}(y = 1|\mathbf{x})$ we can rewrite (2.3) as

$$\frac{\mathrm{p}}{1-\mathrm{p}} = e^{\boldsymbol{\beta}^\mathsf{T}\mathbf{x}},$$

and solving for p we have:

$$\begin{aligned}
\mathrm{p} &= (1-\mathrm{p})e^{\boldsymbol{\beta}^\mathsf{T}\mathbf{x}} \\
\mathrm{p} &= e^{\boldsymbol{\beta}^\mathsf{T}\mathbf{x}} - \mathrm{p}(e^{\boldsymbol{\beta}^\mathsf{T}\mathbf{x}}) \\
\mathrm{p}(1 + e^{\boldsymbol{\beta}^\mathsf{T}\mathbf{x}}) &= e^{\boldsymbol{\beta}^\mathsf{T}\mathbf{x}} \\
\mathrm{p} &= \frac{e^{\boldsymbol{\beta}^\mathsf{T}\mathbf{x}}}{1 + e^{\boldsymbol{\beta}^\mathsf{T}\mathbf{x}}},
\end{aligned}$$

which we can further simplify to:

$$\mathrm{p} = \frac{\frac{e^{\boldsymbol{\beta}^\mathsf{T}\mathbf{x}}}{e^{\boldsymbol{\beta}^\mathsf{T}\mathbf{x}}}}{\frac{1+e^{\boldsymbol{\beta}^\mathsf{T}\mathbf{x}}}{e^{\boldsymbol{\beta}^\mathsf{T}\mathbf{x}}}} = \frac{1}{\frac{1}{e^{\boldsymbol{\beta}^\mathsf{T}\mathbf{x}}} + \frac{e^{\boldsymbol{\beta}^\mathsf{T}\mathbf{x}}}{e^{\boldsymbol{\beta}^\mathsf{T}\mathbf{x}}}} = \frac{1}{\frac{1}{e^{\boldsymbol{\beta}^\mathsf{T}\mathbf{x}}} + 1} = \frac{1}{1 + e^{-\boldsymbol{\beta}^\mathsf{T}\mathbf{x}}}.$$

To summarize, logistic regression is modeling $\mathbb{P}(y = 1|\mathbf{x})$ as $\frac{1}{1+e^{-\boldsymbol{\beta}^\mathsf{T}\mathbf{x}}}$. Hence, we can rewrite our decision function as:

$$\frac{1}{1+e^{-\boldsymbol{\beta}^{\mathsf{T}}\mathbf{x}}} \underset{f(\mathbf{x})\,=\,0}{\overset{f(\mathbf{x})\,=\,1}{\gtrless}} 1 - \frac{1}{1+e^{-\boldsymbol{\beta}^{\mathsf{T}}\mathbf{x}}}, \tag{2.4}$$

which intuitively says: if $\mathbb{P}(y=1|\mathbf{x})$ is larger than $\mathbb{P}(y=0|\mathbf{x})$ (or equivalently, larger than $1/2$), then we conclude that $y=1$, and otherwise we conclude that $y=0$.

This means that if you want to know whether your girlfriend/boyfriend is cheating on you, all you have to do is plug her/his feature vector $\mathbf{x}$ in (2.4), and decide accordingly. The catch here is that (2.4) depends on $\boldsymbol{\beta}$, which you do not know a priori. So, which $\boldsymbol{\beta}$ should you use? The answer is: you have to infer/learn it.

## 2.3 Inferring/Learning $\beta$

Logistic regression uses (2.4) to decide whether $y=0$ or $y=1$. However, our function $f$ in (2.4) depends on $\boldsymbol{\beta}$, which is unknown a priori. To estimate/learn $\boldsymbol{\beta}$ we use *training* data, meaning a collection of pairs $(\mathbf{x}_1,\mathrm{y}_1),(\mathbf{x}_2,\mathrm{y}_2),\dots,(\mathbf{x}_\mathrm{n},\mathrm{y}_\mathrm{n})$ containing features $\mathbf{x}_1,\mathbf{x}_2,\dots,\mathbf{x}_\mathrm{n}$ and their corresponding variables of interest $\mathrm{y}_1,\mathrm{y}_2,\dots,\mathrm{y}_\mathrm{n}$. In our example, this would mean features about n people and their information about whether they are cheating or not.

In words, our goal is to find the parameter $\boldsymbol{\beta}$ that best explains our training samples. More precisely, we want to find the parameter $\boldsymbol{\beta}$ that maximizes the *likelihood* of our sample. Intuitively, this likelihood is the chance that our observed samples are correctly predicted by $f$ (which depends on $\boldsymbol{\beta}$), i.e., the probability that $\mathrm{y}_\mathrm{i} = f(\mathbf{x}_\mathrm{i})$, for every $\mathrm{i}=1,\dots,\mathrm{n}$.

## 2.4 Likelihood

Recall that a probability distribution $\mathbb{P}(x=\mathrm{x}|\theta)$ determines the probability that a random variable $x$ takes a certain value x, given some parameter $\theta$. For example, if $x \sim$ *Bernoulli*(p), then the probability that $x$ takes the value 1 is p. In this case p is the parameter $\theta$.

Similarly, the likelihood $\mathbb{L}(\theta|x=\mathrm{x})$ determines the probability that a parameter $\theta$ was the one that generated a sample x.

**Example 2.1.** Suppose we know $x$ is distributed i.i.d. *Bernoulli*(p), and we observe $\mathrm{x}=1$. Then the likelihood of parameter p given sample $\mathrm{x}=1$ is:

$$\mathbb{L}(\mathrm{p}|\mathrm{x}=1) \;=\; \mathbb{P}(x|\mathrm{p})\Big|_{x=1} \;=\; \mathrm{p}.$$

$\mathbb{P}(x|\theta)$ and $\mathbb{L}(\theta|\mathrm{x})$ may *look* the same. The difference is that $\mathbb{P}(x|\theta)$ is a function where $x$ is the variable, and $\theta$ is fixed. In contrast, $\mathbb{L}(\theta|\mathrm{x})$ is a function where $\theta$ is the variable, and x is fixed. In other words, we use $\mathbb{P}(x|\theta)$ when we have not observed $x$, but we know $\theta$; we use $\mathbb{L}(\theta|\mathrm{x})$ when we have already observed x, and we want to know the likelihood that a certain parameter $\theta$ that generated it. This is why we use of x (as opposed to $x$), to denote that data is already observed.

**Example 2.2.** Suppose $x \sim \mathcal{N}(\mu, 1)$. Then

$$\mathbb{P}(x|\mu) \;=\; \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(x-\mu)^2}.$$

Notice that with probability, $\mu$ is known, and $x$ is the variable. In contrast, with the likelihood, x is known, and $\mu$ is the variable:

$$\mathbb{L}(\mu|\mathrm{x}) \;=\; \mathbb{P}(x|\mu)\Big|_{x=\mathrm{x}} \;=\; \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(\mathrm{x}-\mu)^2}$$

**Example 2.3.** Suppose $x_1, \dots, x_6$ are distributed i.i.d. *Bernoulli*($^1/_4$). Then the probability that $\mathrm{x}_1 = \mathrm{x}_2 = \mathrm{x}_3 = 1$, and $\mathrm{x}_4 = \mathrm{x}_5 = \mathrm{x}_6 = 0$ is:

$$\begin{aligned}\mathbb{P}(x_1 = x_2 = x_3 = 1, x_4 = x_5 = x_6 = 0|\mathrm{p}) \;&=\; \prod_{\mathrm{i}=1}^{3} \mathbb{P}(x_\mathrm{i} = 1|\mathrm{p}) \cdot \prod_{\mathrm{i}=4}^{6} \mathbb{P}(x_\mathrm{i} = 0|\mathrm{p}) \\ &=\; \mathrm{p}^3(1-\mathrm{p})^3 \;=\; (^1/_4)^3\ (^3/_4)^3.\end{aligned}$$

Instead, suppose that we observe $\mathrm{x}_1 = \mathrm{x}_2 = \mathrm{x}_3 = 1$, and $\mathrm{x}_4 = \mathrm{x}_5 = \mathrm{x}_6 = 0$. Then the likelihood of p under this sample is:

$$\mathbb{L}(\mathrm{p}|\mathrm{x}_1, \dots, \mathrm{x}_6) \;=\; \prod_{\mathrm{i}=1}^{6} \mathbb{P}(x_\mathrm{i}|\mathrm{p})\Big|_{x_\mathrm{i}=\mathrm{x}_\mathrm{i}} \;=\; \prod_{\mathrm{i}=1}^{3} \mathbb{P}(x_\mathrm{i}|\mathrm{p})\Big|_{x_\mathrm{i}=1} \cdot \prod_{\mathrm{i}=4}^{6} \mathbb{P}(x_\mathrm{i}|\mathrm{p})\Big|_{x_\mathrm{i}=0} \;=\; \mathrm{p}^3(1-\mathrm{p})^3.$$

Based on this sample, which would be your best guess at the value of p? Is this the same value that maximizes $\mathbb{L}(\mathrm{p}|\mathrm{x}_1, \dots, \mathrm{x}_6)$?

## 2.5 Maximum Likelihood

Back to logistic regression, we can model our training data $\mathrm{y}_1, \dots, \mathrm{y}_\mathrm{n}$ as i.i.d. realizations of a *Bernoulli*(p) random variable, where $\mathrm{p} = \frac{1}{1+e^{-\boldsymbol{\beta}^\mathsf{T}\mathbf{x}_\mathrm{i}}}$. A little thought shows that the likelihood of a *Bernoulli*(p) random variable can be written as:

$$\mathbb{L}(\mathrm{p}|\mathrm{y}) \;=\; \mathrm{p}^\mathrm{y}(1-\mathrm{p})^{1-\mathrm{y}}.$$

Make sure you understand why this is true. By independence, the likelihood of our training sample is:

$$\mathbb{L}(\mathrm{p}|\mathrm{y}_1, \dots, \mathrm{y}_\mathrm{n}) \;=\; \prod_{\mathrm{i}=1}^{\mathrm{n}} \mathbb{L}(\mathrm{p}|\mathrm{y}_\mathrm{i}) \;=\; \prod_{\mathrm{i}=1}^{\mathrm{n}} \mathrm{p}^{\mathrm{y}_\mathrm{i}}(1-\mathrm{p})^{1-\mathrm{y}_\mathrm{i}}.$$

Since p is in turn a function of the unknown parameter $\boldsymbol{\beta}$ and the known data $\mathbf{x}_1, \dots, \mathbf{x}_\mathrm{n}$, we can rewrite this as

$$\mathbb{L}(\boldsymbol{\beta}|\mathrm{y}_1, \dots, \mathrm{y}_\mathrm{n}, \mathbf{x}_1, \dots, \mathbf{x}_\mathrm{n}) \;=\; \prod_{\mathrm{i}=1}^{\mathrm{n}} \left(\frac{1}{1+e^{-\boldsymbol{\beta}^\mathsf{T}\mathbf{x}_\mathrm{i}}}\right)^{\mathrm{y}_\mathrm{i}} \left(1 - \frac{1}{1+e^{-\boldsymbol{\beta}^\mathsf{T}\mathbf{x}_\mathrm{i}}}\right)^{1-\mathrm{y}_\mathrm{i}}. \tag{2.5}$$

To ease our notation we will use $\mathbb{L}(\boldsymbol{\beta}|\mathbf{Y}, \mathbf{X})$ as shorthand for $\mathbb{L}(\boldsymbol{\beta}|\mathrm{y}_1, \dots, \mathrm{y}_\mathrm{n}, \mathbf{x}_1, \dots, \mathbf{x}_\mathrm{n})$. Our goal is to find the $\boldsymbol{\beta}$ that maximizes this likelihood. Maximizing products as in (2.5) can be difficult (as you know from the chain rule of derivatives), so to simplify this maximization, we will use a common trick: apply log, so that products transform into sums, which are easily maximized (because of the linearity of derivatives: the derivative of a sum is the sum of derivatives). We know we can do this because $\mathbb{L}$ is positive (so we can apply log), and log is monotonically increasing, implying that

$$\arg\min_{\boldsymbol{\beta}\in\mathbb{R}^{\mathrm{d}}} \ \mathbb{L}(\boldsymbol{\beta}|\mathbf{Y},\mathbf{X}) \ = \ \arg\min_{\boldsymbol{\beta}\in\mathbb{R}^{\mathrm{d}}} \ \log\left[\mathbb{L}(\boldsymbol{\beta}|\mathbf{Y},\mathbf{X})\right].$$

So instead of maximizing the likelihood directly, we can equivalently maximize the so-called log-likelihood:

$$\begin{aligned}
\ell(\boldsymbol{\beta}|\mathbf{Y},\mathbf{X}) \ &:= \ \log\left[\mathbb{L}(\boldsymbol{\beta}|\mathbf{Y},\mathbf{X})\right] && (2.6)\\
&= \ \log\left[\prod_{\mathrm{i}=1}^{\mathrm{n}}\left(\frac{1}{1+e^{-\boldsymbol{\beta}^\mathsf{T}\mathbf{x}_\mathrm{i}}}\right)^{\mathrm{y}_\mathrm{i}}\left(1-\frac{1}{1+e^{-\boldsymbol{\beta}^\mathsf{T}\mathbf{x}_\mathrm{i}}}\right)^{1-\mathrm{y}_\mathrm{i}}\right] && (2.7)\\
&= \ \sum_{\mathrm{i}=1}^{\mathrm{n}}\log\left[\left(\frac{1}{1+e^{-\boldsymbol{\beta}^\mathsf{T}\mathbf{x}_\mathrm{i}}}\right)^{\mathrm{y}_\mathrm{i}}\left(1-\frac{1}{1+e^{-\boldsymbol{\beta}^\mathsf{T}\mathbf{x}_\mathrm{i}}}\right)^{1-\mathrm{y}_\mathrm{i}}\right] && (2.8)\\
&= \ \sum_{\mathrm{i}=1}^{\mathrm{n}}\mathrm{y}_\mathrm{i}\log\left(\frac{1}{1+e^{-\boldsymbol{\beta}^\mathsf{T}\mathbf{x}_\mathrm{i}}}\right)+(1-\mathrm{y}_\mathrm{i})\log\left(1-\frac{1}{1+e^{-\boldsymbol{\beta}^\mathsf{T}\mathbf{x}_\mathrm{i}}}\right), && (2.9)
\end{aligned}$$

which is easier to maximize than (2.5) because it contains a sum, rather than a product. Sadly, (2.6) is still complex enough that it cannot be maximized with our calculus 101 recipe (take derivative, set to zero, and solve for the optimizer), so instead we will use gradient ascent.

## 2.6 Gradient Ascent

One often wants to find the *maximizer* of a function $g(\beta)$, that is, the value $\beta^\star$ such that $g(\beta^\star) \geq g(\beta)$ for every $\beta$ in the domain of $g$. If $g$ is concave and *simple* enough, $\beta^\star$ can be determined using our elemental calculus recipe:

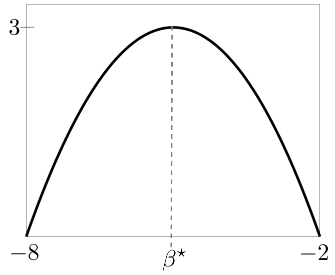
1. Take derivative of $g(\beta)$
2. Set derivative to zero, and solve for the maximizer.

**Example 2.4.** Consider $g(\beta) = 3 - (\beta+5)^2$. We can follow our recipe to find its maximizer:

1. The derivative of $g$ is given by $\nabla g(\beta) = -2(\beta+5)$.
2. Setting the derivative to zero and solving for $\beta$ we obtain:

$$\begin{aligned}
-2(\beta+5) \ &= \ 0\\
\beta \ &= \ -5.
\end{aligned}$$

Since $g$ is concave (can you show this?), we conclude that its maximizer is $\beta^\star = -5$, as depicted below:



Some functions, however, are either not concave, or too complex that we cannot solve for $\beta$ in step 2. For example, the gradient of (2.6) is:

$$\nabla \ell(\boldsymbol{\beta}|\mathbf{Y},\mathbf{X}) \;=\; \sum_{\mathrm{i}=1}^{\mathrm{N}} \left( \mathrm{y_i} - \frac{1}{1+e^{-\boldsymbol{\beta}^\mathsf{T}\mathbf{x}_\mathrm{i}}} \right) \mathbf{x}_\mathrm{i}. \tag{2.10}$$

If we set this to zero, can you solve for $\boldsymbol{\beta}$?

For cases where our calculus 101 recipe does not work, we use *optimization*, which is the field of mathematics that deals with finding maximums (and minimums). In particular, we will use one of the most elemental tools of optimization: gradient ascent.

The setting is is this: you have a function $g(\beta)$. You want to find its maximum. You cannot solve for it directly using the derivative trick, so what can you do? You can *test* the value of $g$ for different values of $\beta$. For example, you can test $g(0)$, then maybe $g(1)$, then maybe $g(-1)$, then maybe $g(1.5)$, and so on, until you find the maximizer. Of course, depending on the domain of $g$, there could be infinitely many options, so testing them all would be infeasible.

As the name suggests, the main idea of gradient ascent is to test some initial value $\beta_0$ (for example 0), and iteratively use the gradient (another name for derivative) to determine which value of $\beta$ to test next, such that the each new value $\beta_{t+1}$ produces a higher value for $g$, until we find the maximum. The main intuition is that the gradient $\nabla g(\beta)$ tells us the slope of $g$ at $\beta$. If this slope is positive, then we know that $g$ is increasing, and we should try a larger value of $\beta$, say $\beta_{t+1} = \beta_t + \eta$, where $\eta$ is often referred to as *step-size*. If the slope is negative, then we know that $g$ is decreasing, and we should try a smaller value of $\beta$, say $\beta_{t+1} = \beta_t - \eta$ (see Figure 2.1 to build some intuition).

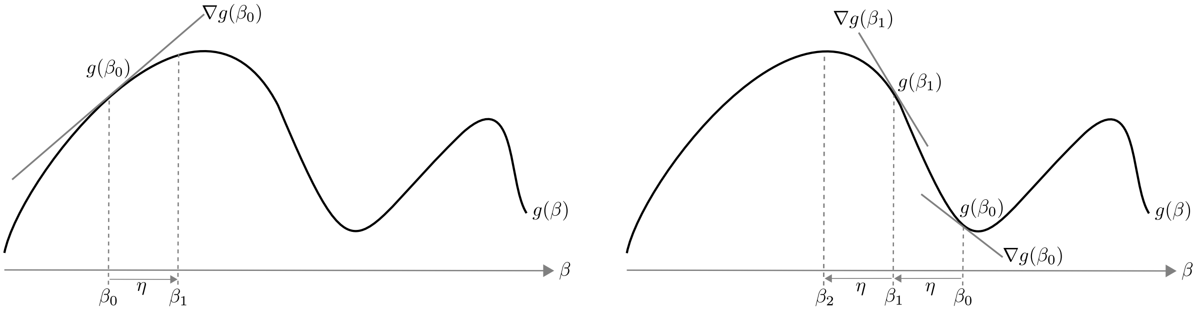


Figure 2.1: Start at some point $\beta_0$. If the gradient is positive (left figure), try a larger value of $\beta$, say $\beta_1 = \beta_0 + \eta$. If the gradient is negative (right figure), try a smaller value of $\beta$, say $\beta_1 = \beta_0 + \eta$. Repeat this until convergence.

The same insight extends to multivariable functions. If $g$ is a function of a vector $\boldsymbol{\beta} \in \mathbb{R}^\mathrm{d}$, then $\nabla g(\boldsymbol{\beta}) \in \mathbb{R}^\mathrm{d}$

gives the slope of $g$ in each of the d coordinates of $\beta$. Based on this insight, gradient ascent can be summarized as follows:

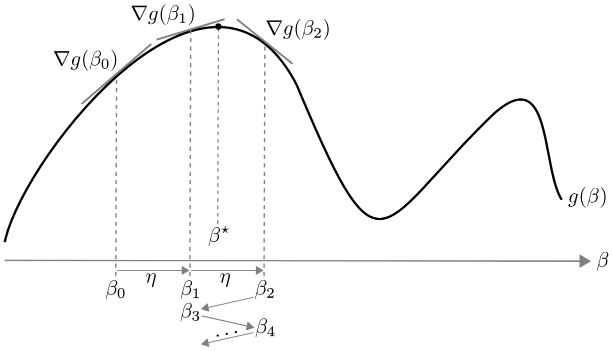
```
Algorithm 1: Gradient Ascent
Input: Function $g$, step-size parameter $\eta > 0$.
Initialize $\boldsymbol{\beta}_0$. For example, $\boldsymbol{\beta}_0 = \mathbf{0}$.
Repeat until convergence: $\boldsymbol{\beta}_{t+1} = \boldsymbol{\beta}_t + \eta \nabla g(\boldsymbol{\beta}_t)$.
Output: $\boldsymbol{\beta}^\star = \boldsymbol{\beta}_t$.
```

### 2.6.1 Step-size $\eta$

The keen reader will be wondering, what if we move too far? In our example of Figure 2.1, we could run into an infinite loop, where

$$\begin{aligned} \beta_1 &= \beta_3 = \beta_5 = \beta_7 = \cdots \\ \beta_2 &= \beta_4 = \beta_6 = \beta_8 = \cdots, \end{aligned}$$

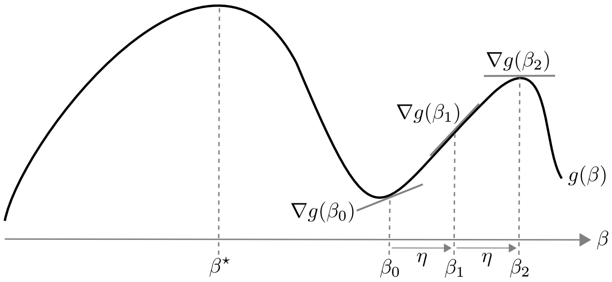
without ever achieving $\beta^\star$, as depicted below:



How would you solve this?

### 2.6.2 Initialization

The keen reader will also be wondering: what if we start at the wrong place, as depicted below:



In cases like these we could run into a so-called local maximum, that is, a point that is larger than all other points in its vicinity, but not necessarily the maximum over the whole domain of $g$. In the figure above, $\beta_2$ is a local maximizer.

How would you solve this?

## 2.7 Maximizing Likelihood for Logistic Regression

Equipped with gradient ascent, we can go back to logistic regression to find the parameter $\boldsymbol{\beta}$ that maximizes the likelihood. All we need to do is use gradient ascent to find:

$$\boldsymbol{\beta}^\star \;=\; \arg\max_{\boldsymbol{\beta}\in\mathbb{R}^{\mathrm{d}}} \; \mathbb{L}(\boldsymbol{\beta}|\mathbf{Y},\mathbf{X}).$$

Once we have found $\boldsymbol{\beta}^\star$, we can determine whether $y = 0$ or $y = 1$ for a new sample with features $\mathbf{x}$, by simply using (2.4), with $\boldsymbol{\beta}^\star$ instead of $\boldsymbol{\beta}$.