Exercises 3.1 **Fitting a regression line to the student debt data**

Fit a linear model to the U.S. student loan debt dataset shown in Fig. 1.8, called *student_debt_data.csv*, by solving the associated linear regression Least Squares problem. If this linear trend continues what will the total student debt be in 2050?

Exercises 3.3 **The Least Squares cost for linear regression is convex**

Show that the Least Squares cost function for linear regression written compactly as in Section 3.1.3,

$$g\left(\tilde{\mathbf{w}}\right) = \sum_{p=1}^{P}\left(\tilde{\mathbf{x}}_p^T\tilde{\mathbf{w}} - y_p\right)^2, \tag{3.31}$$

is a convex quadratic function by completing the following steps.

a) Show that $g\left(\tilde{\mathbf{w}}\right)$ can be written as a quadratic function of the form

$$g\left(\tilde{\mathbf{w}}\right) = \frac{1}{2}\tilde{\mathbf{w}}^T\mathbf{Q}\tilde{\mathbf{w}} + \mathbf{r}^T\tilde{\mathbf{w}} + d \tag{3.32}$$

by determining proper $\mathbf{Q}$, $\mathbf{r}$, and d.

b) Show that $\mathbf{Q}$ has all nonnegative eigenvalues (*hint: see Exercise 2.10*).

c) Verify that $\nabla^2 g\left(\tilde{\mathbf{w}}\right) = \mathbf{Q}$ and so that g satisfies the second order definition of convexity, and is therefore convex.

d) Show that applying a single Newton step (see Section 2.2.4) to minimize the Least Squares cost function leads to precisely the first order system of linear equations discussed in Section 3.1.3, i.e., to the system $\left(\sum_{p=1}^{P}\tilde{\mathbf{x}}_p\tilde{\mathbf{x}}_p^T\right)\tilde{\mathbf{w}} = \sum_{p=1}^{P}\tilde{\mathbf{x}}_p y_p$. (This is because the Least Squares cost is a quadratic function, as in Example 2.7.)

Exercises 3.10 **Logistic regression as a linear system**

In this exercise you will explore particular circumstances that allow one to transform the nonlinear system of equations in (3.24) into a system which is linear in the parameters b and w. In order to do this recall that a function f has an *inverse* at t if another function f^{-1} exists such that $f^{-1}(f(t)) = t$. For example, the exponential function $f(t) = e^t$ has the inverse $f^{-1}(t) = \log(t)$ for every t since we always have $f^{-1}(f(t)) = \log\left(e^t\right) = t$.

a) Show that the logistic sigmoid has an inverse for each t where $0 < t < 1$ of the form $\sigma^{-1}(t) = \log\left(\frac{t}{1-t}\right)$ and check that indeed $\sigma^{-1}(\sigma(t)) = t$ for all such values of t.

b) Suppose for a given dataset $\left\{\left(x_p, y_p\right)\right\}_{p=1}^{P}$ that $0 < y_p < 1$ for all p. Apply the sigmoid inverse to the system shown in Equation (3.24) to derive the equivalent set of linear equations

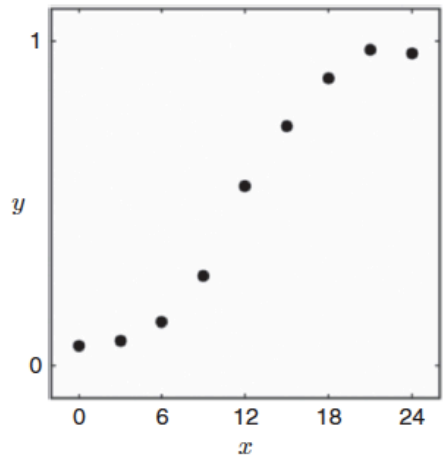
$$b + x_p w \approx \log\left(\frac{y_p}{1 - y_p}\right) \quad p = 1, \ldots, P. \tag{3.36}$$

Since the equations in (3.36) are now linear in both b and w we may solve for these parameters by simply checking the first order condition for optimality.

c) Using the dataset *bacteria_data.csv* solve the Least Squares cost function based on the linear system of equations from part **b)** and plot the data, along with the logistic sigmoid fit to the data as shown in Fig. 3.19.

If a dataset of P points $\left\{\left(x_p, y_p\right)\right\}_{p=1}^{P}$ is roughly distributed like a sigmoid function, then this data satisfies

$$\sigma\left(b + x_p w\right) \approx y_p, \quad p = 1, \ldots, P, \tag{3.24}$$



The normalized cell concentration of Lactobacillus delbrueckii in a constrained laboratory environment over the period of 24 hours. Data in this figure is taken from [48].

Exercises 3.11 **Code up gradient descent for logistic regression**

In this exercise you will reproduce the gradient descent paths shown in Fig. 3.11.

a) Verify that the gradient descent step shown in Equation (3.27) is correct.

Note that this gradient can be written more compactly by denoting $\sigma_p^{k-1} = \sigma\left(\tilde{\mathbf{x}}_p^T\tilde{\mathbf{w}}^{k-1}\right)$, $r_p^{k-1} = 2\left(\sigma_p^{k-1} - y_p\right)\sigma_p^{k-1}\left(1-\sigma_p^{k-1}\right)$ for all $p = 1, ..., P$, and $\mathbf{r}^{k-1} = \begin{bmatrix} r_1^{k-1} & r_2^{k-1} & \cdots & r_P^{k-1} \end{bmatrix}^T$, and stacking the $\tilde{\mathbf{x}}_p$ column-wise into the matrix $\tilde{\mathbf{X}}$. Then the gradient can be written as $\nabla g\left(\tilde{\mathbf{w}}^{k-1}\right) = \tilde{\mathbf{X}}\mathbf{r}^{k-1}$. For programming languages like Python and MATLAB/OCTAVE that have especially efficient implementations of matrix/vector operations this can be much more efficient than explicitly summing over the P points as in Equation (3.27).

b) The surface in this figure was generated via the wrapper *nonconvex_logistic_growth* with the dataset *bacteria_data.csv*, and inside the wrapper you must complete a short gradient descent function to produce the descent paths called

$$[\text{in, out}] = \text{grad\_descent}\left(\tilde{\mathbf{X}}, \mathbf{y}, \tilde{\mathbf{w}}^0\right), \tag{3.37}$$

where "in" and "out" contain the gradient steps $\tilde{\mathbf{w}}^k = \tilde{\mathbf{w}}^{k-1} - \alpha_k \nabla g\left(\tilde{\mathbf{w}}^{k-1}\right)$ taken and corresponding objective value $g\left(\tilde{\mathbf{w}}^k\right)$ respectively, $\tilde{\mathbf{X}}$ is the input data matrix, $\mathbf{y}$ the output values, and $\tilde{\mathbf{w}}^0$ the initial point.

Almost all of this function has already been constructed for you. For example, the step length is fixed at $\alpha_k = 10^{-2}$ for all iterations, etc., and you must only enter the gradient of the associated cost function. Pressing "run" in the editor will run gradient descent and will reproduce Fig. 3.11.

$$\nabla g\left(\tilde{\mathbf{w}}\right) = 2\sum_{p=1}^{P}\left(\sigma\left(\tilde{\mathbf{x}}_p^T\tilde{\mathbf{w}}\right) - y_p\right)\sigma\left(\tilde{\mathbf{x}}_p^T\tilde{\mathbf{w}}\right)\left(1 - \sigma\left(\tilde{\mathbf{x}}_p^T\tilde{\mathbf{w}}\right)\right)\tilde{\mathbf{x}}_p. \tag{3.27}$$

Exercises 3.13 **Code up gradient descent for ℓ_2 regularized logistic regression**

In this exercise you will reproduce Fig. 3.13 by coding up gradient descent to minimize the regularized logistic regression Least Squares cost function shown in Equation (3.29).

a) Verify that the gradient of the cost function can be written as

$$\nabla g\left(\tilde{\mathbf{w}}\right) = 2\sum_{p=1}^{P}\left(\sigma\left(\tilde{\mathbf{x}}_p^T\tilde{\mathbf{w}}\right) - y_p\right)\sigma\left(\tilde{\mathbf{x}}_p^T\tilde{\mathbf{w}}\right)\left(1 - \sigma\left(\tilde{\mathbf{x}}_p^T\tilde{\mathbf{w}}\right)\right)\tilde{\mathbf{x}}_p + 2\lambda\begin{bmatrix} 0 \\ \mathbf{w} \end{bmatrix}. \tag{3.40}$$

b) The surface in this figure was generated via the wrapper *l2reg_nonconvex_logistic_growth* with the dataset *bacteria_data.csv*, and inside the wrapper you must complete a short gradient descent function to produce the descent paths called

$$[\text{in, out}] = \text{grad\_descent}\left(\widetilde{\mathbf{X}}, \mathbf{y}, \tilde{\mathbf{w}}^0\right), \tag{3.41}$$

where "in" and "out" contain the gradient steps $\tilde{\mathbf{w}}^k = \tilde{\mathbf{w}}^{k-1} - \alpha_k \nabla g\left(\tilde{\mathbf{w}}^{k-1}\right)$ taken and corresponding objective value $g\left(\tilde{\mathbf{w}}^k\right)$ respectively, $\widetilde{\mathbf{X}}$ is the input data matrix whose pth column is the input data $\tilde{\mathbf{x}}_p$, $\mathbf{y}$ the output values stacked into a column vector, and $\tilde{\mathbf{w}}^0$ the initial point.

Almost all of this function has already been constructed for you. For example, the step length is fixed at $\alpha_k = 10^{-2}$ for all iterations, etc., and you must only enter the gradient of the associated cost function. Pressing "run" in the editor will run gradient descent and will reproduce Fig. 3.13.

$$g\left(b, \mathbf{w}\right) = \sum_{p=1}^{P} \left(\sigma\left(b + \mathbf{x}_p^T \mathbf{w}\right) - y_p\right)^2 + \lambda \left\|\mathbf{w}\right\|_2^2. \tag{3.29}$$