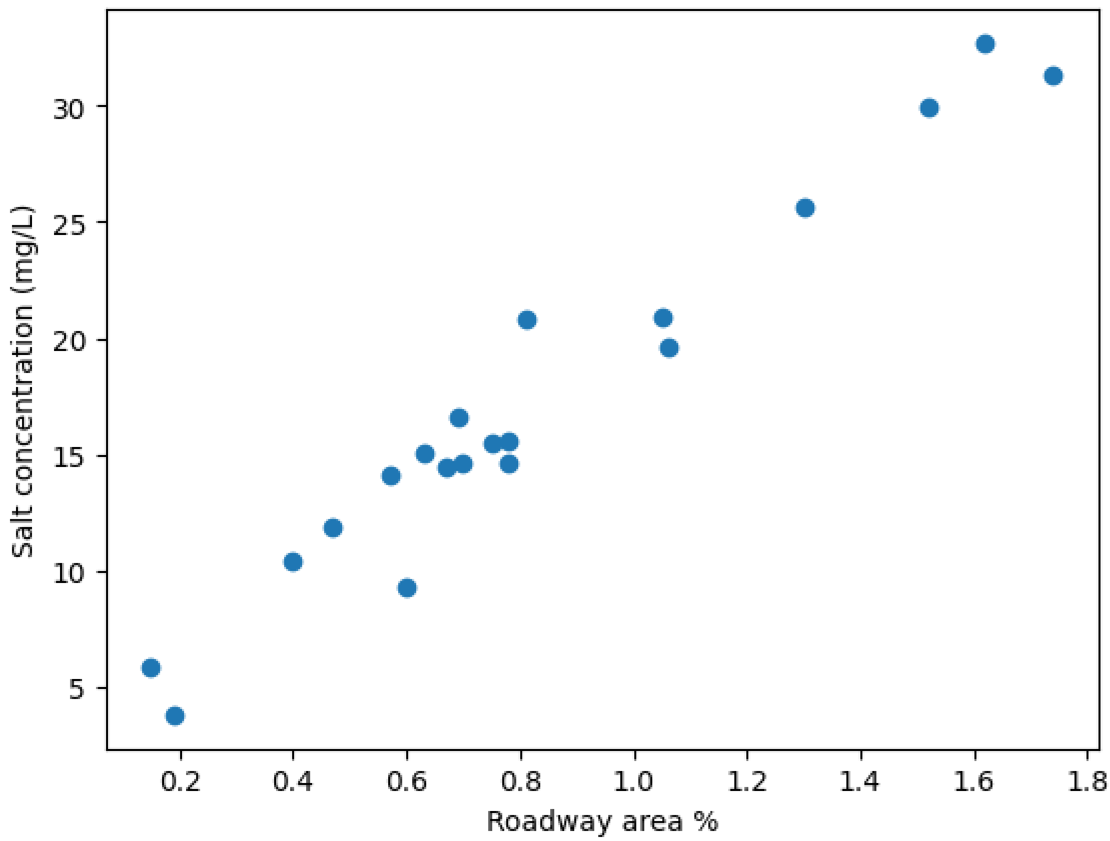


Least squares regression

The problem of linear regression is to find a line that "best fits" the given data. That is we want all the points $\{(x_1, y_1), \ldots, (x_n, y_n)\}$ to satisfy the equation of the line $y = mx + c$. Since we know that there exists no such line, so we will try to make $y \approxeq mx + c$, by minimizing some error/distance/cost/loss function between y and $mx + c$ for every point (x_i, y_i) in the dataset. The simplest error function that results in nice answers is squared distance:

$$e(x_i, y_i) = (y_i - (mx_i + c))^2$$

Then we can minimize the total error to find the line:

$$m^*, c^* = \arg\min_{m,c} \sum_{i=1}^{n} e(x_i, y_i)$$

Geometrically, this error minimization corresponds to minimizing the stubs in the following figure:

Vectorization of Least square regression

y = Salt concentration = Variable to predict = Output

x = Road area = Input variable

$$D = \{(x_1, y_1), (x_2, y_2), \vdots, (x_n, y_n)\}$$

Training Dataset

Model

$$\hat{y} = f(x) = \underbrace{mx + c}_{(m,c)} = \underbrace{w_1 x + w_0}_{(w_1, w_0)}$$

are called parameters / weights

Loss function

Least square regression

$$l(y_i, \hat{y}_i) = (y_i - \hat{y}_i)^2$$

Objective

$$\underset{(m,c)}{\text{minimize}} \sum_{i=1}^{n} l(y_i, \hat{y}_i)$$

$$m^*, c^* = \underset{(m,c)}{\arg\min} \sum_{i=1}^{n} l(y_i, \hat{y}_i)$$

model

$$f(x; (m, c)) = mx + c$$

$$= \underbrace{[m, c]}_{\underline{m}} \underbrace{\begin{bmatrix} x \\ 1 \end{bmatrix}}_{\underline{x}}$$

$$\underline{m}^T \in \mathbb{R}^{1\times 2}$$

$$\underline{m} \in \mathbb{R}^{2\times 1}$$

$$f(\underline{x}; \underline{m}) = \underline{m}^T \underline{x}$$

$$\underline{m}^T \underline{x}$$

$$= \underline{m} \cdot \underline{x}$$

$$= \underline{x} \cdot \underline{m} = \underline{x}^T \underline{m}$$

$$l(y_i, \hat{y}_i) = (y_i - \hat{y}_i)^2$$

$$= (y_i - f(\underline{x}_i; \underline{m}))^2$$

$$= (y_i - \underline{m}^T \underline{x}_i)^2$$

Parameters are same for all datapoints

objective $\sum_{i=1}^{n} l(y_i, \hat{y}_i)$

$$= \sum_{i=1}^{n} (\underbrace{y_i - \underline{m}^T x_i}_{})^2$$

$$e_i := y_i - \underline{m}^T \underline{x}_i$$

$$= \sum_{i=1}^{n} e_i^2$$

$$= \left(\sqrt{\sum_{i=1}^{n} e_i^2}\right)^2$$

$$= \|\underline{e}\|^2$$

$$= \underline{e}^T\underline{e} = \underline{e}\cdot\underline{e}$$

$$\underline{e} := \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_n \end{bmatrix}$$

$$\underline{e} := \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_n \end{bmatrix} = \begin{bmatrix} y_1 - \underline{m}^T\underline{x}_1 \\ y_2 - \underline{m}^T\underline{x}_2 \\ \vdots \\ y_n - \underline{m}^T\underline{x}_n \end{bmatrix} = \underbrace{\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}}_{\underline{y}} - \underbrace{\begin{bmatrix} \underline{m}^T\underline{x}_1 \\ \underline{m}^T\underline{x}_2 \\ \vdots \\ \underline{m}^T\underline{x}_n \end{bmatrix}}$$

$$\begin{bmatrix} \underline{m}^T\underline{x}_1 \\ \underline{m}^T\underline{x}_2 \end{bmatrix} = \begin{bmatrix} [m \;\; c]\begin{bmatrix} x_1 \\ 1 \end{bmatrix} \\ [m \;\; c]\begin{bmatrix} x_2 \\ 1 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} m x_1 + c \\ m x_2 + c \end{bmatrix}$$

$$\begin{bmatrix} \underline{x}_1^T\underline{m} \\ \underline{x}_2^T\underline{m} \end{bmatrix}$$

$$\cancel{[m \;\; c]\begin{bmatrix} x_1 \\ 1 \\ x_2 \\ 1 \end{bmatrix}}$$

$$\begin{bmatrix} x_1 & 1 \\ x_2 & 1 \end{bmatrix}_{2\times 2}\begin{bmatrix} m \\ c \end{bmatrix}_{2\times 1}$$

$$\begin{bmatrix} \underline{x}_{1\,1\times 2}^T \\ \underline{x}_{2\,1\times 2}^T \end{bmatrix}_{2\times 2}\underline{m}_{2\times 1}$$

$$\cancel{\underline{m}^T_{1\times 2}\begin{bmatrix} \underline{x}_{1\,2\times 1} \\ \underline{x}_{2\,2\times 1} \end{bmatrix}_{4\times 1}}$$

$$A_{m\times n}B_{n\times p}$$

$$\underline{e} := \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_n \end{bmatrix} = \begin{bmatrix} y_1 - \underline{m}^T \underline{x}_1 \\ y_2 - \underline{m}^T \underline{x}_2 \\ \vdots \\ y_n - \underline{m}^T \underline{x}_n \end{bmatrix} = \underbrace{\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}}_{\underline{y}} - \underbrace{\begin{bmatrix} \underline{m}^T \underline{x}_1 \\ \underline{m}^T \underline{x}_2 \\ \vdots \\ \underline{m}^T \underline{x}_n \end{bmatrix}}$$

$$= \underline{y}_{n\times 1} - \underbrace{\begin{bmatrix} \underline{x}_1^T \\ \underline{x}_2^T \\ \vdots \\ \underline{x}_n^T \end{bmatrix}}_{X}{}_{n\times 2}\, \underline{m} \qquad = \underline{y}_{n\times 1} - X_{n\times 2}\, \underline{m}$$

$$= \underline{e}$$

$$X := \begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ x_3 & 1 \\ \vdots & \\ x_n & 1 \end{bmatrix}_{n\times 2}$$

Objective

$$\underset{\substack{(m,c) \\ (\underline{m})}}{\text{minimize}} \|\underline{e}\|^2 = \underline{e}^T \underline{e} = (\underline{y}_{n\times 1} - X_{n\times 2}\, \underline{m}_{2\times 1})^T (\underline{y} - X\underline{m})$$

$$(A+B)^T \overset{?}{=} A^T + B^T \qquad (-B)^T = B^T$$

$$(A-B)^T = A^T - B^T$$

$$\left(\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} + \begin{bmatrix} b_{11} & b_{12} \\ b_{22} & b_{22} \end{bmatrix} \right)^T$$

$$= \left(\begin{pmatrix} a_{11}+b_{11} & a_{12}+b_{12} \\ a_{21}+b_{21} & a_{22}+b_{22} \end{pmatrix} \right)^T$$

$$= \begin{pmatrix} a_{11}+b_{11} & a_{21}+b_{21} \\ a_{12}+b_{12} & a_{22}+b_{22} \end{pmatrix}$$

$$= \begin{bmatrix} a_{11} & a_{21} \\ a_{12} & a_{22} \end{bmatrix} + \begin{pmatrix} b_{11} & b_{21} \\ b_{12} & b_{22} \end{pmatrix}$$

$$= A^T + B^T$$

$$(AB)^T = B^T A^T$$

---

objective fn $\quad (\underline{y} - X\underline{m})^T (\underline{y} - X\underline{m})$

$$= \left[\underline{y}^T - (X\underline{m})^T \right] (\underline{y} - X\underline{m})$$

$$= \left(\underline{y}^T - \underset{1\times 2}{\underline{m}^T}\ \underset{2\times n}{X^T}\right)\left(\underline{y} - X\underline{m}\right)$$

$$= \underline{y}^T(\underline{y} - X\underline{m}) - \underline{m}^T X^T(\underline{y} - X\underline{m})$$

$$= \underbrace{\underset{1\times n}{\underline{y}^T}\ \underset{n\times 1}{\underline{y}}}_{\underset{1\times 1}{\in \mathbb{R}}} - \underbrace{\underset{1\times n}{\underline{y}^T}\ \underset{n\times 2}{X}\ \underset{2\times 1}{\underline{m}}}_{\underset{1\times 1}{\in \mathbb{R}}}{}^{①} - \underbrace{\underline{m}^T X^T \underline{y}}_{\in \mathbb{R}}{}^{②} + \underbrace{\underline{m}^T X^T X \underline{m}}_{\in \mathbb{R}}$$

$$A \in \mathbb{R}^{1\times 1} \qquad A = [a_{11}]$$

$$A^T = [a_{11}]^T = [a_{11}] = A$$

$$(\underline{y}^T X \underline{m})^T = \underline{y}^T X \underline{m} \qquad (AB)^T = B^T A^T$$

$$\underline{m}^T X^T \underline{y}^{②} = y^T X \underline{m}^{①} \qquad (A^T)^T = A$$

objective fn

$$\underbrace{R(\underline{m}; X, \underline{y})}_{\text{empirical risk}} = \underline{y}^T \underline{y} - 2 y^T X \underline{m} + \underline{m}^T X^T X \underline{m}$$

Quadrati function $= a m^2 + b m + c$

for 2 var in scalar form

$$= \underbrace{ax^2 + by^2}+ \underbrace{cx + dy}_{\to [c\ d]\begin{bmatrix} x \\ y \end{bmatrix}} + \underbrace{exy} + f$$

$$\underbrace{[x\ \ y]}_{\underline{x}^T} \underbrace{\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} x \\ y \end{bmatrix}}_{\underline{x}} = \underline{x}^T A \underline{x}$$

$$= [x\ \ y] \begin{bmatrix} a_{11} x + a_{12} y \\ a_{21} x + a_{22} y \end{bmatrix}$$

$$= a_{11} x^2 + \underbrace{a_{12} xy + a_{21} xy} + \underbrace{a_{22} y^2}$$

Quadratic function in vector form is

$$= \underline{x}^T A \underline{x} + \underline{b}^T \underline{x} + c$$

compare it to the scalar form $ax^2 + by^2 + cx + dy + exy + f$

$$= [x\ \ y] \begin{bmatrix} a & e/2 \\ e/2 & b \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + [c\ \ d] \begin{bmatrix} x \\ y \end{bmatrix} + f$$

Objective function

$$R(\underline{m}; X, y) = \underbrace{\underline{y}^T \underline{y}}_{\text{constant}} - \underbrace{2 \underbrace{y^T X}_{1\times n} \underline{m}}_{\to\ \underline{b}^T_{1\times 2} \underline{m}}{}_{n\times 2} + \underline{m}^T \underbrace{X^T X}_{2\times n\ \ n\times 2 \ \to\ \underline{m}^T A \underline{m}} \underline{m}$$

$R(\underline{m}; X, \underline{y})$ is a quadratic polynomial in $\underline{m}$

$$\left.\frac{\partial}{\partial \underline{m}} R(\underline{m}; X, \underline{y})\right|_{\underline{m}^*} = 0$$

Derivative of a vector-valued function w.r.t a vector

$$\underline{f}(\underline{x}) \rightarrow \underline{f} \qquad \underline{f} : \mathbb{R}^n \longrightarrow \mathbb{R}^m$$

$$\underline{x} \in \mathbb{R}^n \qquad \underline{f} \in \mathbb{R}^m$$

$$\boxed{\frac{\partial}{\partial \underline{x}} \underline{f}(\underline{x})} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \cdots & & \vdots \\ \vdots & \cdots & \frac{\partial f_i}{\partial x_j} & \vdots \\ \frac{\partial f_m}{\partial x_1} & & & \frac{\partial f_m}{\partial x_n} \end{bmatrix}_{m \times n} \qquad \underline{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \qquad \underline{f} = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_m \end{bmatrix}$$

Jacobian

$$\boxed{\nabla^T_{\underline{x}} f(\underline{x})} = \frac{\partial}{\partial \underline{x}} f(\underline{x}) = \left[\frac{\partial f(\underline{x})}{\partial x_1} \ \cdots \ \frac{\partial f(\underline{x})}{\partial x_n} \right]$$

Gradient transpose

$$f(\underline{x}) = \underline{x}^T A \underline{x} + \underline{b}^T \underline{x} + c \qquad \underline{x} \in \mathbb{R}^n$$

$$\frac{\partial}{\partial \underline{x}} f(\underline{x}) \qquad \frac{\partial}{\partial \underline{x}} c = [0, 0, \ldots 0]_{1\times n}$$

$$= \underline{0}^T_{1\times n}$$

$$\underline{b}^T \underline{x} = [b_1 \; b_2 \; \ldots \; b_n] \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = b_1 x_1 + b_2 x_2 + \ldots + b_n x_n$$

$$\frac{\partial}{\partial \underline{x}} \underline{b}^T \underline{x} = \left[\frac{\partial}{\partial x_1} \underline{b}^T \underline{x} \; \ldots \; \frac{\partial}{\partial x_n} \underline{b}^T \underline{x} \right]$$

$$= [b_1 \; b_2 \; \ldots \; b_n] = \underline{b}^T$$

$$\frac{\partial}{\partial \underline{x}} \underline{x}^T A \underline{x} = \underline{x}^T (A + A^T)$$

$$f(\underline{x}) = \underline{x}^T A \underline{x} + \underline{b}^T \underline{x} + c$$

$$\frac{\partial}{\partial \underline{x}} f(\underline{x}) = x^T (A + A^T) + b^T = \underline{0}^T$$

$$\boxed{\underline{x} = \underbrace{(A + A^T)}_{\text{if invertible}}{}^{-1} \underline{b}}$$

$$R(\underline{m};X,\underline{y}) = \underbrace{\underline{y}^T\underline{y}}_{\text{constant}} - \underbrace{2\underbrace{\underline{y}^T}_{1\times n}\underbrace{X}_{n\times 2}\underline{m}}_{\underline{b}^T_{1\times 2}\,\underline{m}} + \underbrace{\underline{m}^T \underbrace{X^T}_{2\times n}\underbrace{X}_{n\times 2}\underline{m}}_{\underline{m}^T A\, \underline{m}}$$

$$\frac{\partial}{\partial \underline{m}} R(\underline{m}) = 0^T$$

$$-2y^T X + \underline{m}^T\left(X^TX + (X^TX)^T\right) = \underline{0}^T$$

$$\Rightarrow -2\underline{y}^T X + 2\underline{m}^T(X^TX) = \underline{0}^T$$

$$\Rightarrow \underline{m}^T(X^TX) = \underline{y}^T X$$

$$\Rightarrow \underline{m}^T\underbrace{(X^TX)(X^TX)^{-1}} = y^T X(X^TX)^{-1}$$

Multiple to the righ $(X^TX)^{-1}$ on both sides

$$m^T = y^T X(X^TX)^{-1}$$

$$\boxed{\underline{m} = (X^TX)^{-1}X^T\underline{y}}$$

$$\begin{bmatrix} m \\ c \end{bmatrix} = \left(\underbrace{\begin{bmatrix} x_1 & x_2 & \cdots & x_n \\ 1 & 1 & \cdots & 1 \end{bmatrix}}_{2\times n}\begin{pmatrix} x_1 & 1 \\ x_2 & 1 \\ \vdots & \\ x_n & 1 \end{pmatrix}\right)^{-1}_{2\times 2} \underbrace{\begin{pmatrix} x_1 & \cdots & x_n \\ 1 & & 1 \end{pmatrix}}_{2\times n}\underbrace{\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}}_{n\times 1}$$

2×1

Recall that the magnitude of a vector $\|\mathbf{v}\| = \sqrt{v_1^2 + v_2^2 + \cdots + v_n^n}$ has a similar form to the error function. This suggests that we can define an error vector with the signed error for each data point as it's elements

$$\mathbf{e} = \begin{bmatrix} y_1 - (mx_1 + c) \\ y_2 - (mx_2 + c) \\ \vdots \\ y_n - (mx_n + c) \end{bmatrix}$$

Minimizing the total error is same as minimizing the square of error vector magnitude

$$m^*, c^* = \arg \min_{m,c} \|\mathbf{e}\|^2$$

While we are at at it let us define $\mathbf{x} = [x_1; \dots; x_n]$ to denote the vector of all x coordinates of the dataset and $\mathbf{y} = [y_1; \dots; y_n]$ to denote y coordinates. Then the error vector is:

$$\mathbf{e} = \mathbf{y} - (\mathbf{x}m + \mathbf{1}_n c)$$

where $\mathbf{1}_n$ is a n-D vector of all ones. Finally, we vectorize parameters of the line $\mathbf{m} = [m; c]$. We will also need to horizontally concatenate $\mathbf{x}$ and $\mathbf{1}_n$. Let's call the result $\mathbf{X} = [\mathbf{x}, \mathbf{1}_n] \in \mathbb{R}^{n \times 2}$. Now, the error vector looks like this:

$$\mathbf{e} = \mathbf{y} - \mathbf{X}\mathbf{m}$$

Expanding the error magnitude:

$$\|\mathbf{e}\|^2 = (\mathbf{y} - \mathbf{X}\mathbf{m})^\top (\mathbf{y} - \mathbf{X}\mathbf{m})$$
$$= \mathbf{y}^\top \mathbf{y} + \mathbf{m}^\top \mathbf{X}^\top \mathbf{X}\mathbf{m} - 2\mathbf{y}^\top \mathbf{X}\mathbf{m}$$

Homework 3: Problem 4

Expand

$$(\mathbf{y} - \mathbf{X}\mathbf{m})^\top (\mathbf{y} - \mathbf{X}\mathbf{m})$$

and show that it is equal to

$$\mathbf{y}^\top \mathbf{y} + \mathbf{m}^\top \mathbf{X}^\top \mathbf{X}\mathbf{m} - 2\mathbf{y}^\top \mathbf{X}\mathbf{m}$$

Our minimization problem in vectorized form is:

$$\mathbf{m}^* = \arg\min_{\mathbf{m}} \mathbf{y}^\top\mathbf{y} + \mathbf{m}^\top\mathbf{X}^\top\mathbf{X}\mathbf{m} - 2\mathbf{y}^\top\mathbf{X}\mathbf{m}$$

This is a quadratic equation in $\mathbf{m}$ that can be minimized by equating the derivate to zero.

Two rules of vector derivatives

There are two conventions in vector derivatives:

1. Gradient convention
2. Jacobian convention

Gradient convention

Under gradient convention the derivative of scalar-valued vector function function $f(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}$ is defined as vertical stacking of element-wise derivatives

$$\frac{\partial}{\partial \mathbf{x}} f(\mathbf{x}) = \begin{bmatrix} \frac{\partial f(\mathbf{x})}{\partial x_1} \\ \vdots \\ \frac{\partial f(\mathbf{x})}{\partial x_n} \end{bmatrix} \in \mathbb{R}^n$$

Jacobian convention

Under gradient convention the derivative of scalar-valued vector function function $f(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}$ is defined as horizontal stacking of element-wise derivatives

$$\frac{\partial}{\partial \mathbf{x}} f(\mathbf{x}) = \begin{bmatrix} \frac{\partial f(\mathbf{x})}{\partial x_1} & \dots & \frac{\partial f(\mathbf{x})}{\partial x_n} \end{bmatrix} \in \mathbb{R}^{1\times n}$$

For a vector-value vector function $\mathbf{f}(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}^m$, Jacobian of $\mathbf{f}(\mathbf{x})$ is the vertical concatentation of gradients transposed, resulting in $m \times n$ matrix

$$\mathbf{J}_{\mathbf{x}}(\mathbf{f}(\mathbf{x})) = \frac{\partial}{\partial \mathbf{x}} \mathbf{f}(\mathbf{x}) = \begin{bmatrix} \frac{\partial f_1(\mathbf{x})}{\partial \mathbf{x}} \\ \dots \\ \frac{\partial f_m(\mathbf{x})}{\partial \mathbf{x}} \end{bmatrix}$$

We will use Jacobian convention in this course, because it works nicely with chain rule.

Derivative of a linear function

All scalar-valued linear functions of $\mathbf{x}$ can be written in the form $f(\mathbf{x}) = \color{red}{\backslash\mathrm{bfc}}^\top \mathbf{x}$.

$$\frac{\partial}{\partial \mathbf{x}} \mathbf{c}^\top \mathbf{x} = \mathbf{c}^\top \tag{10}$$

Derivative of a quadratic function

All scalar-valued homogeneous quadratic functions of $\mathbf{x}$ can be written in the form $f(\mathbf{x}) = \mathbf{x}^\top \mathbf{A} \mathbf{x}$.

$$\frac{\partial}{\partial \mathbf{x}} \mathbf{x}^\top \mathbf{A} \mathbf{x} = \mathbf{x}^\top (\mathbf{A} + \mathbf{A}^\top) \tag{11}$$

Homework 3: Problem 5

Proof of above two derivatives is left as an exercises.

Back to Least square regression

$$\mathbf{0}^\top = \frac{\partial}{\partial \mathbf{m}} (\mathbf{y}^\top \mathbf{y} + \mathbf{m}^\top \mathbf{X}^\top \mathbf{X} \mathbf{m} - 2\mathbf{y}^\top \mathbf{X} \mathbf{m}) \tag{12}$$

$$= 2\mathbf{m}^{*\top} \mathbf{X}^\top \mathbf{X} - 2\mathbf{y}^\top \mathbf{X} \tag{13}$$

This gives us the solution

$$\mathbf{m}^* = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}$$

The symbol $\mathbf{V}^{-1}$ is called inverse of matrix $\mathbf{V}$.

The term $(\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top$ is also called the pseudo-inverse of a matrix $\mathbf{X}$, denoted as $\mathbf{X}^\dagger$.

In [26]:
```
n = salt_concentration_data.shape[0]
bfx = salt_concentration_data[:, 2:3]
bfy = salt_concentration_data[:, 1]
bfX = np.hstack((bfx, np.ones((bfx.shape[0], 1))))
bfX
```

Out[26]:
```
array([[0.19, 1.  ],
       [0.15, 1.  ],
       [0.57, 1.  ],
       [0.4 , 1.  ],
       [0.7 , 1.  ],
       [0.67, 1.  ],
       [0.63, 1.  ],
       [0.47, 1.  ],
       [0.75, 1.  ],
       [0.6 , 1.  ],
       [0.78, 1.  ],
       [0.81, 1.  ],
       [0.78, 1.  ],
       [0.69, 1.  ],
       [1.3 , 1.  ],
       [1.05, 1.  ],
       [1.52, 1.  ],
       [1.06, 1.  ],
       [1.74, 1.  ],
       [1.62, 1.  ]])
```

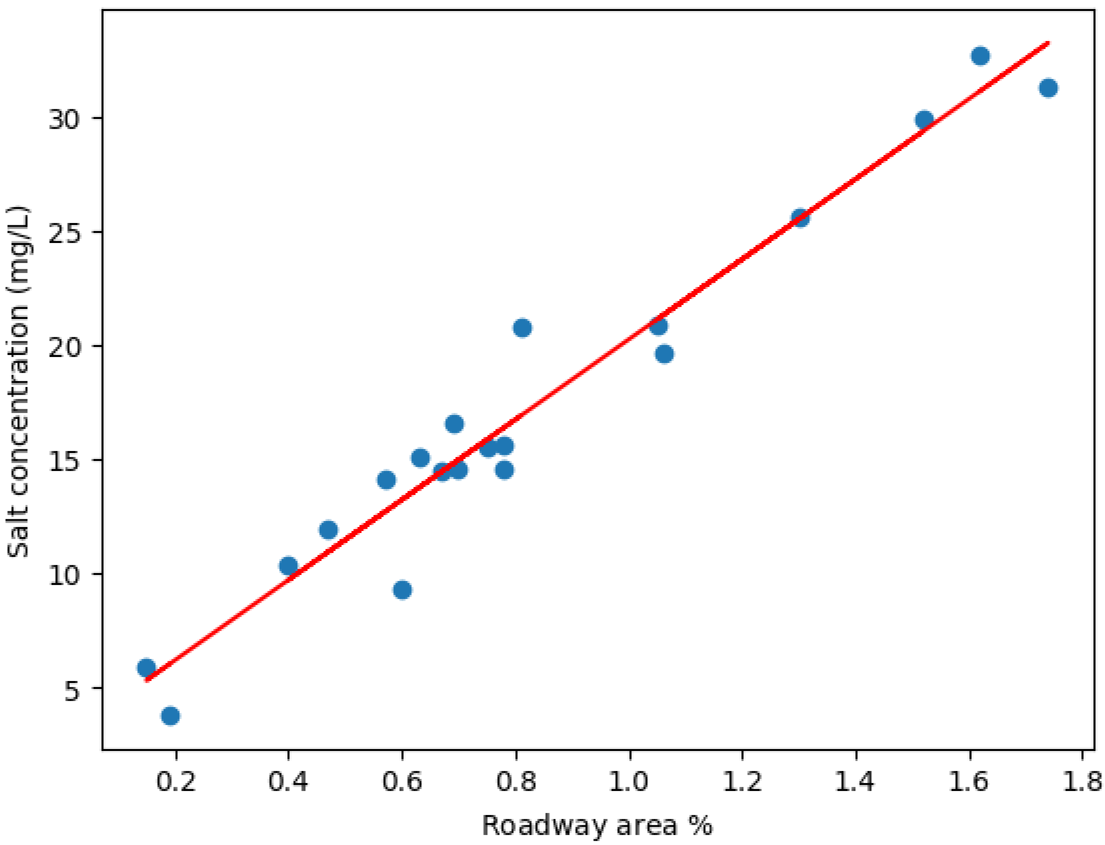
In [27]:
```
bfm = np.linalg.inv(bfX.T @ bfX) @ bfX.T @ bfy
print(bfm)
bfm, *_ = np.linalg.lstsq(bfX, bfy, rcond=None)
print(bfm)
```

```
[17.5466671   2.67654631]
[17.5466671   2.67654631]
```

In [28]:
```
m = bfm.flatten()[0]
c = bfm.flatten()[1]

# Plot the points
fig, ax = plt.subplots()
ax.scatter(salt_concentration_data[:, 2], salt_concentration_data[:, 1])
ax.set_xlabel(r"Roadway area $\%$")
ax.set_ylabel(r"Salt concentration (mg/L)")
x = salt_concentration_data[:, 2]
y = m * x + c
# Plot the points
ax.plot(x, y, 'r-') # the line
```

Out[28]:
```
[<matplotlib.lines.Line2D at 0x7fd3ed57a4a0>]
```


Salt concentration (mg/L)
Roadway area %
5
10
15
20
25
30
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6
1.8