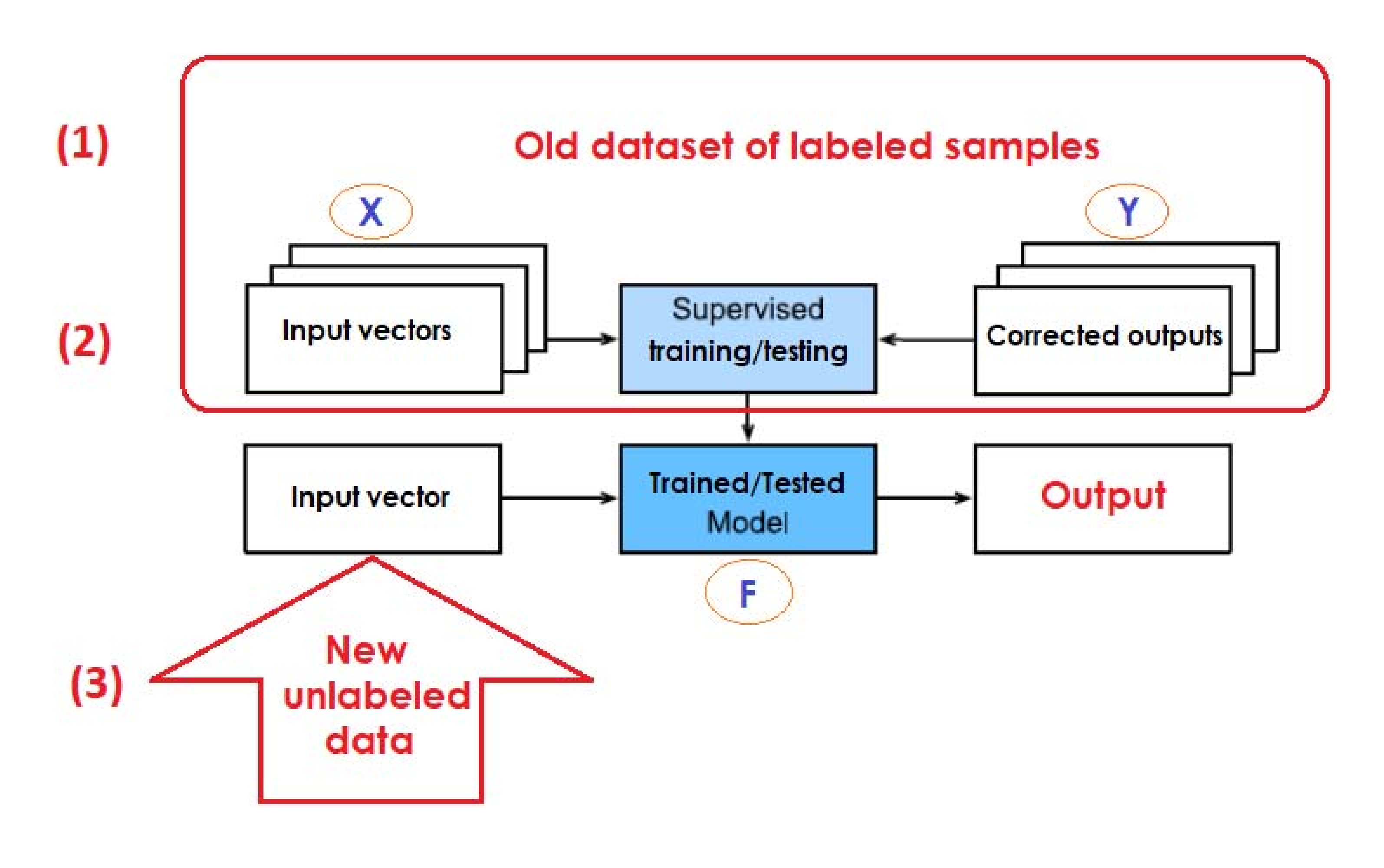
Regression and Classification

Predictive modelling

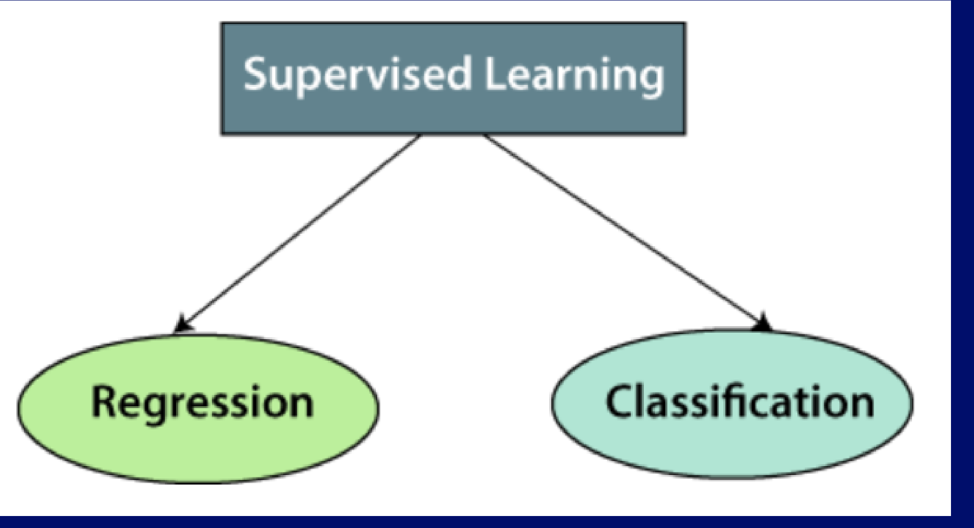
- Developing a model using historical data to make a prediction on new data where we do not have the answer.
- Approximating a mapping function (f) from input variables (x) to output variables (y). This is called the problem of function approximation.

$$y = f(x) + \varepsilon$$

$\Rightarrow$ the modeling algorithm needed to find the best mapping function considering the time and resources available.



Predictive modelling from labeled dataset and supervised algorithm



<u>Output</u>:
Continuous numeric

<u>Method</u>:
Linear regression
ANN

<u>Output</u>:
Discrete categorical

<u>Method</u>:
Logistic regression
ANN

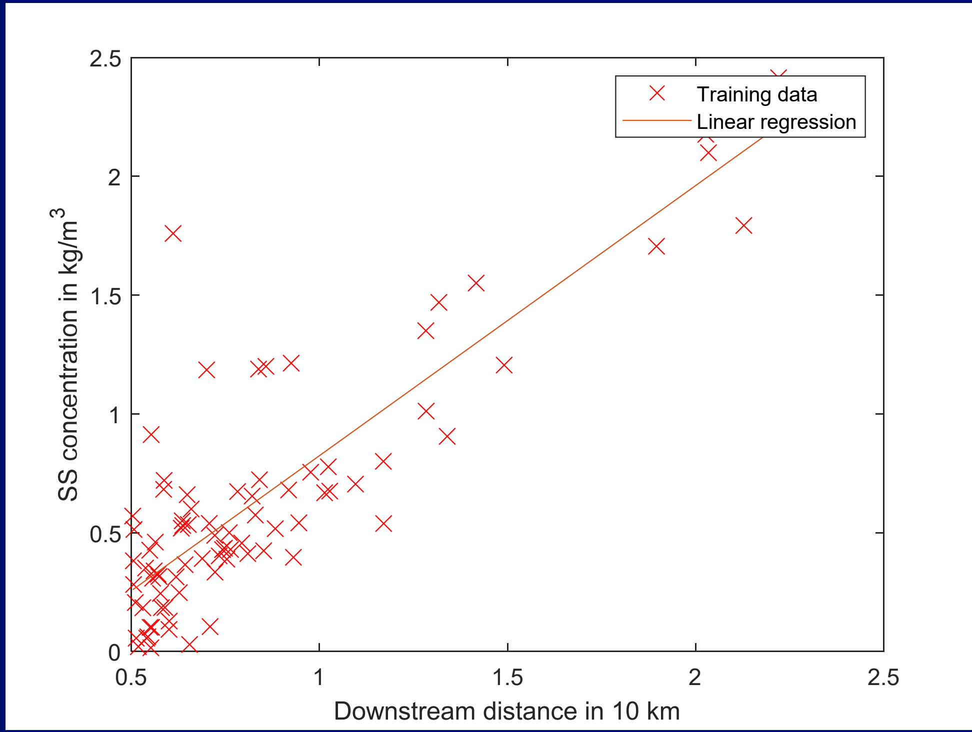
Linear regression

- Linear relationship between the input variables and the output variable
  - Simple linear regression $f(x) = \theta_0 + \theta_1 x$
  - Multiple linear regression $f(\vec{x}) = \theta_0 + \theta_1 x_1 + \ldots + \theta_n x_n$

Simple linear regression

- Example: measured SS concentration along a river reach
- Assuming a linear relationship between concentration (*y*) and downstream distance (*x*):

$$f_\theta(x) = \theta_0 + \theta_1 x$$
$$y = f_\theta(x) + \varepsilon$$

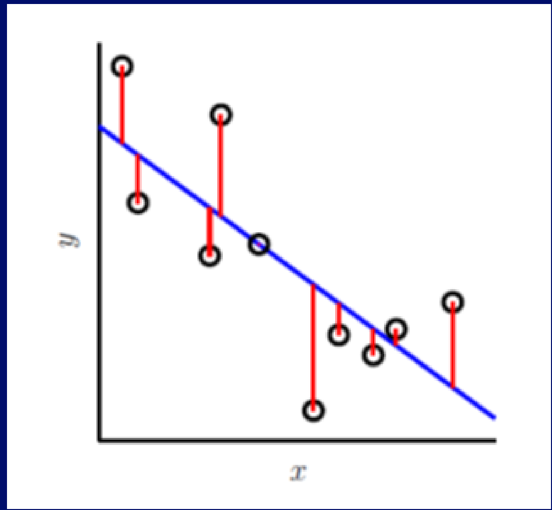


- θ_0 - the intercept term (predicted value of *y* when *x* = 0)
- θ_1 - the slope (the average increase in *y* associated with a one-unit increase in *x*)

- The error term is a catch-all for what we miss with this simple model:
  - the true relationship is probably not linear,
  - there may be other variables that cause variation in *y*, and
  - there may be measurement error.

We typically assume that the error term is independent of *x*.

Least square applied to measure the fitting of the hypothesis function on *m* examples

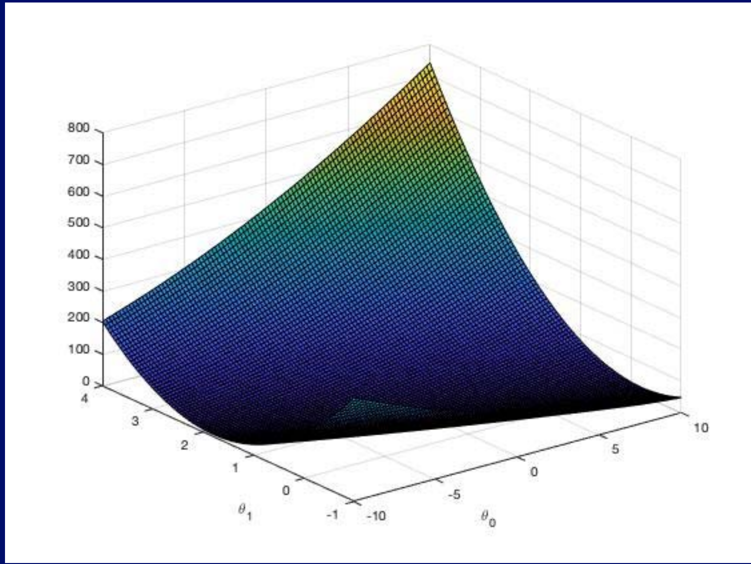


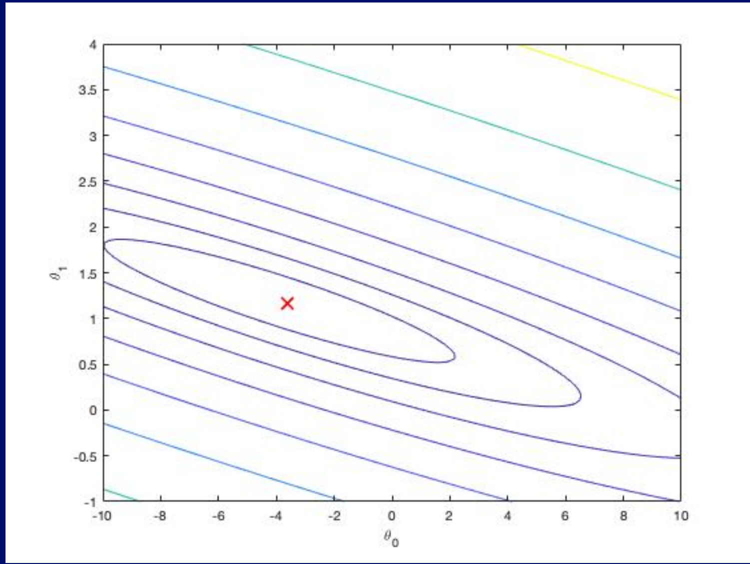
$$error^{(i)} = f_\theta\left(X^{(i)}\right) - Y^{(i)}$$

$$J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}\left(error^{(i)}\right)^2 = \frac{1}{2m}\sum_{i=1}^{m}\left(\theta_0 + \theta_1 x_1^{(i)} - Y^{(i)}\right)^2$$

- $J(\theta)$ - the Cost function
- Objective of Linear regression is to minimize $J(\theta)$ by adjusting θ

Surface and contours of J(θ)





- Gradient of J(θ)

$$\frac{\partial J}{\partial \theta_0} = \frac{1}{m}\sum_{i=1}^{m}\left(\theta_0 + \theta_1 x_1^{(i)} - Y^{(i)}\right)$$

$$\frac{\partial J}{\partial \theta_1} = \frac{1}{m}\sum_{i=1}^{m}\left(\theta_0 + \theta_1 x_1^{(i)} - Y^{(i)}\right)x_1^{(i)}$$

- Gradient descent

$$\theta_j := \theta_j - \alpha \frac{\partial J}{\partial \theta_0}$$

- We store each example as a row in a MATLAB matrix:
  - o X and Y are (m×1) column vectors
  - o θ is (2×1) column vector.
- To rewrite the hypothesis function, cost function and its gradients, and the gradient descent expression in matrix forms, we add an additional first column to X and set it to all ones.

$$f_\theta(x) = X\theta$$

$$error = \left(X\theta - Y\right)$$

$$J\left(\theta\right) = \frac{1}{2m}\left(X\theta - Y\right)'\left(X\theta - Y\right)$$

$$\nabla J\left(\theta\right) = \frac{1}{m}X'\left(X\theta - Y\right)$$

$$\theta := \theta - \alpha \nabla J\left(\theta\right)$$

Exercise 3.1:

Predict suspended sediment concentration at locations of 5 and 15 km downstream.

- Implement linear regression with one variable to predict suspended sediment concentration along a river reach.
- Data set: *example_3_1.txt* (with 84 examples):
  - x_1 - Downstream distance in 10 km
  - *Y* - SS concentration in kg/m^3.
- Uncompleted MATLAB-scripts
  - *Exercise3_1.m* - Script that steps you through this exercise
  - *PlotData.m* - Function to display the dataset
  - *ComputeCost.m* - Function to compute the cost of linear regression
  - *GradientDescent.m* - Function to run gradient descent
- Your task: to complete these MATLAB-scripts and run the program.

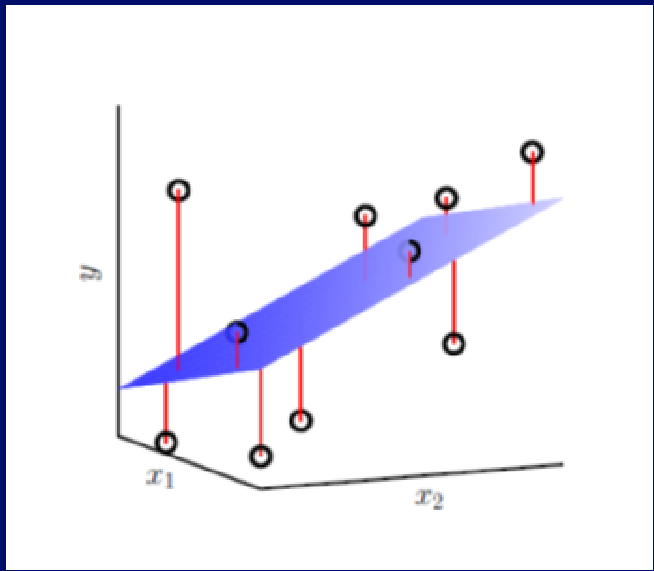
Multiple linear regression

- Linear regression with multiple input variables
- Assuming a linear relationship between output (*Y*) and inputs (*X*)

$$f_\theta(X) = \theta_0 + \theta_1 x_1 + \ldots + \theta_n x_n$$

$$error^{(i)} = f_\theta\left(X^{(i)}\right) - Y^{(i)}$$

$$J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}\left(error^{(i)}\right)^2$$



- J(θ) - the Cost function
- Minimize J(θ) by adjusting θ

- We store each example as a row in a MATLAB matrix:
  - X is (m×n) matrix.
  - Y is (m×1) column vector.
  - θ is (n×1) column vector.
- Adding an additional first column to X and set it to all ones, we can rewrite the hypothesis function, cost function and its gradients, and the gradient descent expression in matrix forms.

$$f_\theta(x) = X\theta$$

$$error = \left(X\theta - Y\right)$$

$$J\left(\theta\right) = \frac{1}{2m}\left(X\theta - Y\right)'\left(X\theta - Y\right)$$

$$\nabla J\left(\theta\right) = \frac{1}{m}X'\left(X\theta - Y\right)$$

$$\Rightarrow$$

$$\theta := \theta - \alpha\nabla J\left(\theta\right)$$

Feature Normalization

- Needed when feature magnitudes with large differences.
- First performing feature scaling can make gradient descent converge much more quickly.

$$x_{j,norm} = \left(x_j - \mu_j\right) / \sigma_j$$

- o μ_j - mean value of each feature
- o σ_j - standard deviation of each feature

```
Loading data ...
First 10 examples from the dataset:
 x = [2104 3], y = 399900
 x = [1600 3], y = 329900
 x = [2400 3], y = 369000
 x = [1416 2], y = 232000
 x = [3000 4], y = 539900
 x = [1985 4], y = 299900
 x = [1534 3], y = 314900
 x = [1427 3], y = 198999
 x = [1380 3], y = 212000
 x = [1494 3], y = 242500
```

Normal equation

- The closed-form solution to linear regression:

$$\nabla J(\theta) = \frac{1}{m} X'(X\theta - Y) = 0$$

  – (X'X) convertible: $\theta = (X'X)^{-1} X'Y$

  – (X'X) nonconvertible: $\theta \approx pinv(X'X) X'Y$

- The Tab Key helps to input the MATLAB-functions names
- The semicolon symbol at the end of a line suppresses the screen output.

Homework 3: **(deadline for submission is 24th May 2022)**

Predict housing price for a house of 1330 m^2 with 3 bedrooms in the city .

- Implement linear regression with two variables to predict housing prices in a city.
- Data set: *example_HW3.txt* (with 47 examples):
  - x_1 - size of the house (in m^2)
  - x_2 - number of bed rooms
  - Y - price in USD.
- Uncompleted MATLAB-scripts
  - *Homework3.m* - Script that steps you through this exercise
  - *PlotData.m* - Function to display the dataset
  - *ComputeCostMulti.m* - Function to compute the cost for multiple variables
  - *GradientDescentMulti.m* - Function to run gradient descent
  - *FeatureNomalize.m* - Function to normalize features
  - *NormalEqn.m* - Function to compute the normal equations

- Your task: 1. to complete these MATLAB-scripts and run the program.

  2. Define the hypothesis function using the gradient descent method and the normal equation