ECE 469/ECE 568 Machine Learning

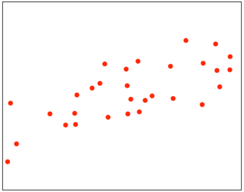
Textbook:
Machine Learning: a Probabilistic Perspective by Kevin Patrick Murphy

Southern Illinois University

August 28, 2024

Regression or Polynomial Curve Fitting Problem

- Suppose we observe a real-valued input variable x.
- We wish to use this observation to predict the value of a real-valued output/target variable y.
- Suppose that we are given a training set consisting of N observations of $\mathbf{x}$: denoted by $\mathbf{x} = [x_1, \cdots, x_N]^T$ and the corresponding observations of the values of $\mathbf{y}$: denoted by $\mathbf{y} = [y_1, \cdots, y_N]^T$.
- Our goal is to utilize this training set to make predictions of the value $\hat{y}$ of the output/target variable for some new value $\hat{x}$ of the input variable.



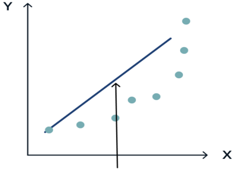
x	y
0.86	2.49
0.09	0.83
-0.85	-0.25
0.87	3.10
-0.44	0.87
-0.43	0.02
-1.10	-0.12
0.40	1.81
-0.96	-0.83
0.17	0.43

Regression or Polynomial Curve Fitting Problem

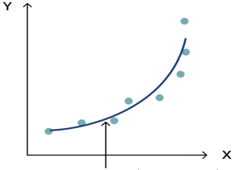
- For example, consider that we have one input feature, denoted by x.
- We shall fit the training data using a polynomial function of the form:

$$y(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \cdots + w_M x^M = \sum_{j=0}^{M} w_j x^j$$

  - M is the order of the polynomial.
  - x^j denotes x raised to the power of j.
  - The polynomial coefficients $w_0, \cdots, w_M$ are collectively denoted by the vector $\mathbf{w} = [w_0, w_1, w_2, \cdots w_M]^T$.
  - Although the polynomial function $y(x, \mathbf{w})$ is a nonlinear function of x, it is a linear function of the coefficients $\mathbf{w}$.



$y = w_0 + w_1 x$



$y = w_0 + w_1 x + w_2 x^2$

A linear hypothesis class

- Consider that we have n input features, denoted by $\mathbf{x} = [x_1, x_2, \cdots, x_n]^T$ in every example.
- Suppose the output variable y is a linear function of x denote mathematically by

$$y = f(\mathbf{x}, \mathbf{w}) = w_0 + w_1 x_1 + \cdots + w_n x_n$$

- Here, w_i are called parameters or weights.
- For notation simplicity, we can let the attribute $x_0 = 1$ (a.k.a. the bias term or intercept term).

$$f(\mathbf{x}, \mathbf{w}) = \sum_{i=1}^{n} w_i x_i = \mathbf{w}^T \mathbf{x}$$

  where $\mathbf{x} = [1, x_1, x_2, \cdots, x_n]^T$ and $\mathbf{w} = [w_0, w_1, w_2, \cdots, w_n]^T$.
- Here, $\mathbf{w}$ and $\mathbf{x}$ are column vectors of size $n + 1$.
- How should we choose parameters/weights $\mathbf{w} = [w_0, w_1, \cdots, w_n]^T$ provided that we are given with a training data set.

How should we choose $\mathbf{w}$?

- We can pick $\mathbf{w}$ to minimize error –> Error minimization!
- Intuitively, $\mathbf{w}$ should make the predictions output variables close to the true values through the learning function/hypothesis f.
- Thus, we need define an error function or cost function to measure how much our prediction differs from the true answer.
- We will choose $\mathbf{w}$ such that an error function is minimized.

How should we choose the error function?

How should we choose $\mathbf{w}$?

- We determine the values of the weights by fitting the polynomial to the training data.
- We accomplish this by minimizing an error function that measures the misfit between the function f, for any given value of $\mathbf{w}$, and the training set data points.
- We can define the sum of squares of errors between the predictions $f(\mathbf{x}, \mathbf{w})$ for each data point x_n and the corresponding target values y_n as an error function.
- Thus, our main idea is to try to make $f(\mathbf{x}, \mathbf{w})$ close to $\mathbf{y}$ on the examples in the training set.
- We can define a sum-of-squares error function:

$$J(\mathbf{w}) = \frac{1}{2} \sum_{i=0}^{m} (f(\mathbf{x}_i, \mathbf{w}) - y_i)^2$$

- Here, $1/2$ is just for convenience.
- We will choose $\mathbf{w}$ such as to minimize $J(\mathbf{w})$.

Minimizing error/cost function

- The cost function minimization can be formulated as

$$\min_{\mathbf{w}=[w_0, w_1, \cdots, w_n]} J(\mathbf{w})$$

- Here, the cost function $J(\mathbf{w})$ is given by

$$J(\mathbf{w}) = \frac{1}{2}\sum_{i=0}^{m}(f(\mathbf{x}_i, \mathbf{w}) - y_i)^2$$

- Thus, we need to optimize parameters/weights $\mathbf{w} = [w_0, w_1, \cdots, w_n]$ to minimize $J(\mathbf{w})$.
- There are two techniques that we can use to find optimal weights/parameters $\mathbf{w}^*$.
  - Analytical solution - can be found for a very few cases/applications
  - Numerical solution - more common approach when the learning model is complicated with many features in the data-set.

Linear regression - Analytical Approach

- By computing the partial derivatives of the cost function $J(\mathbf{w})$ with respect to w_j for $j = \{0, \cdots, n\}$ and equating those to zero, the optimal weights can be found to minimize the cost function.

$$
\begin{aligned}
\frac{\partial}{\partial w_j} J(\mathbf{w}) &= \frac{\partial}{\partial w_j} \left(\frac{1}{2} \sum_{i=0}^{m} (f(\mathbf{x}_i, \mathbf{w}) - y_i)^2 \right) \\
&= \frac{1}{2} \cdot 2 \sum_{i=1}^{m} (f(\mathbf{x}_i, \mathbf{w}) - y_i) \frac{\partial}{\partial w_j} (f(\mathbf{x}_i, \mathbf{w}) - y_i) \\
&= \sum_{i=1}^{m} (f(\mathbf{x}_i, \mathbf{w}) - y_i) \frac{\partial}{\partial w_j} \left(\sum_{l=0}^{n} w_l x_{i,l} - y_i \right) \\
&= \sum_{i=1}^{m} (f(\mathbf{x}_i, \mathbf{w}) - y_i) \cdot x_{i,j}
\end{aligned}
$$

- By setting all these partial derivatives to 0, we get a linear system with $(n+1)$ equations and $(n+1)$ unknowns.
- Via these $(n+1)$ equations, we can solve for $(n+1)$ unknowns to find the optimal weight vector $\mathbf{w}^* = [w_0^*, w_1^*, \cdots, w_n^*]^T$.

Linear regression - Analytical Approach

- If we have a linear model, then we can find a closed-form solution for optimal weights.
- Consider the following linear model

$$f(\mathbf{x}, \mathbf{w}) = \sum_{i=1}^{n} w_i x_i = \mathbf{w}^T \mathbf{x}$$

  where $\mathbf{x} = [1, x_1, x_2, \cdots, x_n]^T$ and $\mathbf{w} = [w_0, w_1, w_2, \cdots, w_n]^T$.

- By using multivariate calculus, a general solution for the optimal weight vector, which minimizes the cost function, can be derived as

$$\boxed{\mathbf{w}^* = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}}$$

  where $\mathbf{X}$ is the input feature matrix concatenated with vectors $\mathbf{x}$ and augmented with a column of ones. Moreover, $\mathbf{y}$ is the column vector of target/output variables.

Linear regression - Analytical Approach

- This solution is typically known as the "nominal equations".
- This is a rare case in which an analytical, exact solution is possible (i.e., the model must be linear).
- You may call a procedure in a linear algebra library for very fast computations.
- This analytical method becomes extremely slow when the training data set grows larger because it needs to compute the inverse $(\mathbf{X}^T\mathbf{X})^{-1}$, which scales as $O(N^2)$ for $N \times N$ matrix $\mathbf{X}$.
- For modern ML applications with extremely large data sets, this analytical solution may not be practically viable.

Linear regression - Analytical Approach

- For example, consider the following training data set.

x_0	x_1	x_2	y
1	1.1	2.2	3.6
1	1.4	1.9	2.4
1	1.0	2.1	1.4

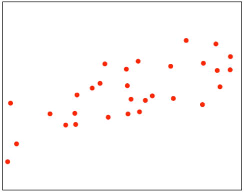
- Then, the optimal weight vector that fits best for the linear model $f(\mathbf{x}, \mathbf{w}) = \sum_{i=1}^{n} w_i x_i = \mathbf{w}^T \mathbf{x}$ with minimal cost function $J(\mathbf{w}) = \frac{1}{2} \sum_{i=0}^{m} (f(\mathbf{x}_i, \mathbf{w}) - y_i)^2$ is $\mathbf{w}^* = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$.
- Here, we can extract $\mathbf{X}$ and $\mathbf{y}$ from the training date set as follows:

$$\mathbf{X} = \begin{bmatrix} 1 & 1.1 & 2.2 \\ 1 & 1.4 & 1.9 \\ 1 & 1.0 & 1.4 \end{bmatrix} \quad \text{and} \quad \mathbf{y} = \begin{bmatrix} 3.6 \\ 2.4 \\ 1.4 \end{bmatrix}$$

- The model that we trained can be written as

$$f(\mathbf{x}, \mathbf{w}^*) = \sum_{i=1}^{n} w_i^* x_i = (\mathbf{w}^*)^T \mathbf{x}$$

Minimizing the cost function - Numerical approach



x	y
0.86	2.49
0.09	0.83
-0.85	-0.25
0.87	3.10
-0.44	0.87
-0.43	0.02
-1.10	-0.12
0.40	1.81
-0.96	-0.83
0.17	0.43

- For illustration purposes, let us choose the simplest linear function with "one parameter" as

$$f(x, \mathbf{w}) = w_1 x, \quad \text{where} \quad \mathbf{w} = [w_1]$$

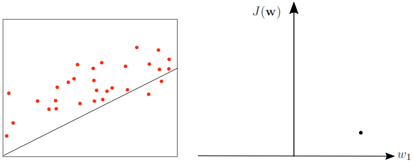
- Here, we need to optimize (find the best value) just one variable w_1 to minimize the cost function $J(w_1)$.

$$\min_{\mathbf{w}=[w_1]} \left(J(\mathbf{w}) = \frac{1}{2} \sum_{i=0}^{m} (f(\mathbf{x}_i, \mathbf{w}) - y_i)^2 = \frac{1}{2} \sum_{i=0}^{m} (w_1 x_i - y_i)^2 \right)$$

Minimizing the cost function - Numerical approach

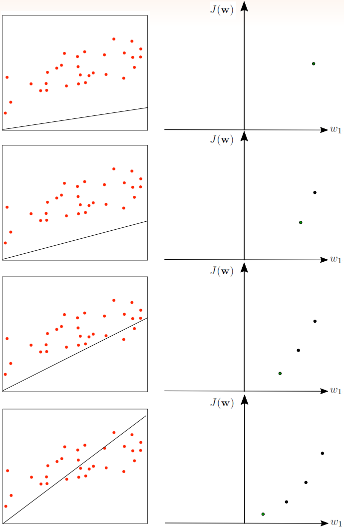
- Thus, we need to numerically find the optimal value for w_1 such that we minimize the cost function $J(w_1)$

$$\min_{w_1} \left(J(w_1) = \frac{1}{2} \sum_{i=0}^{m} (w_1 x_i - y_i)^2 \right)$$



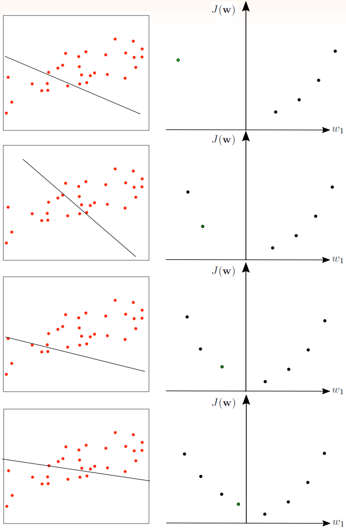
- For a particular choice of w_1, we can compute the cost function $J(w_1)$ and plot it as shown in the figure.
- But, we still do not know whether this choice is the optimal one.
- We need to try out many other choices for w_1 and choose the one that minimizes the cost function.

Minimizing the cost function - Numerical approach



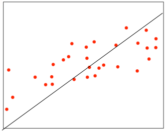
- So far we have tried out positive values for w_1.
- Let us also try some negative values for w_1.

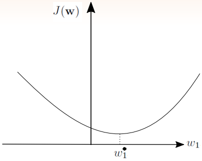
Minimizing the cost function - Numerical approach



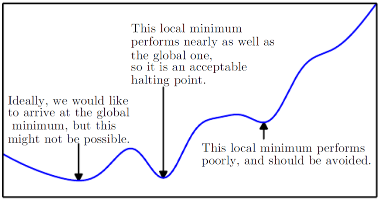
- Similarly, we can choose various values for w_1 and plot the corresponding cost function $J(w_1)$ value.

Minimizing the cost function - Numerical approach





- We can choose the value of w_1 which minimizes the cost function as per this figure.
- The above cost junction is convex and hence it has a global minimum.
- If the function is not convex, then there can be many local minima together with a global minimum.



Appendix: Minimizing the cost function - Analytical Approach

- If all partial derivatives of a function f exist at a point $\mathbf{x} \in \mathbb{R}^n$, then the gradient of f at $\mathbf{x}$ is defined to be the column vector that contains all the partial derivatives.

$$\text{Gradient} = \nabla f(\mathbf{x}) = \begin{bmatrix} \frac{\partial f(\mathbf{x})}{\partial x_1} \\ \frac{\partial f(\mathbf{x})}{\partial x_2} \\ \vdots \\ \frac{\partial f(\mathbf{x})}{\partial x_n} \end{bmatrix} = \begin{bmatrix} \frac{\partial f(\mathbf{x})}{\partial x_1} & \frac{\partial f(\mathbf{x})}{\partial x_2} & \cdots & \frac{\partial f(\mathbf{x})}{\partial x_n} \end{bmatrix}^T$$

- The gradient of our cost function $J(\mathbf{w})$ can be defined similarly as $\nabla J(\mathbf{w}) = \begin{bmatrix} \frac{\partial J(\mathbf{w})}{\partial w_0} & \frac{\partial J(\mathbf{w})}{\partial w_1} & \cdots & \frac{\partial J(\mathbf{w})}{\partial w_n} \end{bmatrix}^T$.

Appendix: Minimizing the cost function - Analytical Approach

- To minimize the cost function with respect to $\mathbf{w}$, we need $\nabla J(\mathbf{w}) = \mathbf{0}$ or $\frac{\partial}{\partial w_j} J(\mathbf{w}) = 0$ for all j.
- We can compute the partial derivatives as follows:

$$
\begin{aligned}
\frac{\partial}{\partial w_j} J(\mathbf{w}) &= \frac{\partial}{\partial w_j} \left(\frac{1}{2} \sum_{i=0}^{m} (f(\mathbf{x}_i, \mathbf{w}) - y_i)^2 \right) \\
&= \frac{1}{2} \cdot 2 \sum_{i=1}^{m} (f(\mathbf{x}_i, \mathbf{w}) - y_i) \frac{\partial}{\partial w_j} (f(\mathbf{x}_i, \mathbf{w}) - y_i) \\
&= \sum_{i=1}^{m} (f(\mathbf{x}_i, \mathbf{w}) - y_i) \frac{\partial}{\partial w_j} \left(\sum_{l=0}^{n} w_l x_{i,l} - y_i \right) \\
&= \sum_{i=1}^{m} (f(\mathbf{x}_i, \mathbf{w}) - y_i) \cdot x_{i,j}
\end{aligned}
$$

By setting all these partial derivatives to 0, we get a linear system with $(n+1)$ equations and $(n+1)$ unknowns.

Appendix: Minimizing cost function - Analytical solution

- let us the following linear model

$$f(\mathbf{x}, \mathbf{w}) = \sum_{i=1}^{n} w_i x_i = \mathbf{w}^T \mathbf{x}$$

where $\mathbf{x} = [1, x_1, x_2, \cdots, x_n]^T$ and $\mathbf{w} = [w_0, w_1, w_2, \cdots, w_n]^T$.

- The associated cost function can be written as

$$||\mathbf{X}\mathbf{w} - \mathbf{y}||^2.$$

where $||\dot{|}||^2$ represents the square norm of a vector: $||\mathbf{a}||^2 = \mathbf{a}^T \mathbf{a}$.

- We would like to minimize this cost function as

$$\min_{\mathbf{w}} ||\mathbf{X}\mathbf{w} - \mathbf{y}||^2.$$

- This is a well-known least-square problem, and the optimal solution for $\mathbf{w}$ has been known for centuries.

Appendix: Minimizing cost function - Analytical solution

- A general closed-form solution can also be obtained by computing the gradient of the cost function and equating it to zero as follows:

  - Recalling some multivariate calculus:

$$
\begin{aligned}
\nabla_{\mathbf{w}} J &= \nabla_{\mathbf{w}} (\mathbf{X}\mathbf{w} - \mathbf{y})^T (\mathbf{X}\mathbf{w} - \mathbf{y}) \\
&= \nabla_{\mathbf{w}} (\mathbf{w}^T \mathbf{X}^T \mathbf{X}\mathbf{w} - \mathbf{y}^T \mathbf{X}\mathbf{w} - \mathbf{w}^T \mathbf{X}^T \mathbf{y} + \mathbf{y}^T \mathbf{y}) \\
&= 2\mathbf{X}^T \mathbf{X}\mathbf{w} - 2\mathbf{X}^T \mathbf{y}
\end{aligned}
$$

  - Setting gradient equal to zero:

$$
\begin{aligned}
2\mathbf{X}^T \mathbf{X}\mathbf{w} - 2\mathbf{X}^T \mathbf{y} &= 0 \\
\Rightarrow \mathbf{X}^T \mathbf{X}\mathbf{w} &= \mathbf{X}^T \mathbf{y}
\end{aligned}
$$

$$
\Rightarrow \mathbf{w} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}
$$

  - The inverse exists if the columns of $\mathbf{X}$ are linearly independent.

This solution is known as "nominal equations".

Appendix: Minimizing cost function - Analytical solution

- Here, we have used the following multi-variate calculus identities in deriving the results in the previous slide.

$$
\begin{aligned}
\nabla_w(\mathbf{w}^T\mathbf{A}\mathbf{w}) &= 2\mathbf{A}\mathbf{w} \\
\nabla_w(\mathbf{w}^T\mathbf{B}\mathbf{y}) &= \mathbf{B}\mathbf{y} \\
\nabla_w(\mathbf{y}^T\mathbf{B}\mathbf{w}) &= \mathbf{B}\mathbf{y}
\end{aligned}
$$

Appendix: Summary - Linear regression - Analytical Approach

- The general solution is $\mathbf{w} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$, where $\mathbf{X}$ is the data matrix augmented with a column of ones, and $\mathbf{y}$ is the column vector of target/output variables.
- This optimal solution for minimizing sum-squared-error can be computed in polynomial time in the size of the data set.
- This is a rare case in which an analytical, exact solution is possible.
- Linear models are an example of parametric models, because we choose a priori a number of parameters that does not depend on the size of the data
- Non-parametric models grow with the size of the data.