Regression

AI/ML Tutorial-3

Yi Huang

Table of Content

Notebook Description

- Notebook 1.1: univariate linear regression
- Notebook 1.2: multivariate linear regression
- Notebook 2: logistic regression
- Notebook 3: regularization
- Notebook 4: from regression to neural network

What is Machine Learning?

According to Wikipedia, **Machine learning** is the study of computer algorithms that can improve automatically through experience and by the use of data.

- Algorithm improves **automatically** according a set of **updating rules**;
- **Data** drives the update, not **human**;
- Human set the updating rules (no worry, machines are not taking over the world here, not yet).

Garbage (Data) in Garbage (algorithm) Out

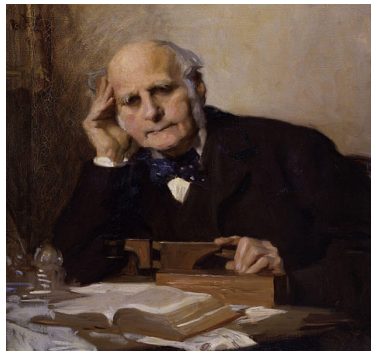
Major types of ML:

- Supervised learning (classification, regression)
  "I know the input and output. Please, machine, find the mapping between them."
- Unsupervised learning (clustering)
  "I know nothing about data. Please, machine, sort them out by similarities."
- Reinforcement learning
  "I know consequences. Please, machine, learn to act properly to induce desirable consequences."

What is Regression

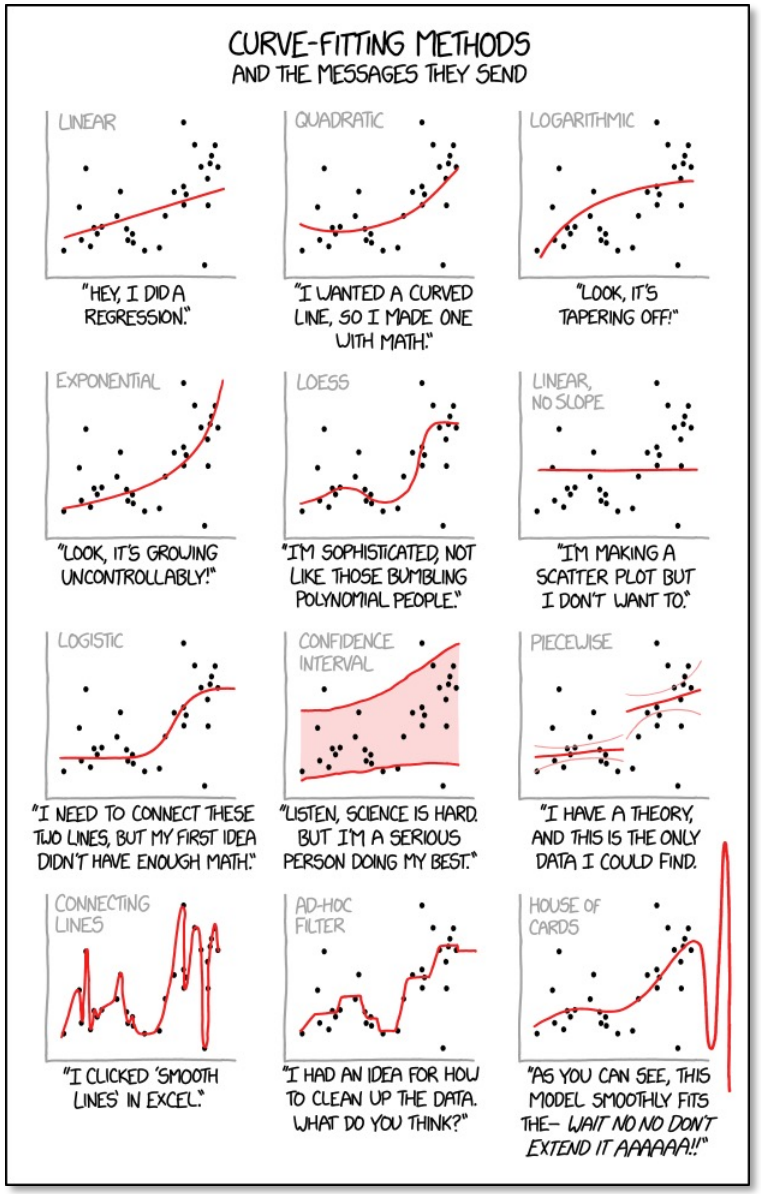
Why regression is call regression: (more fun reading here)

- In medicine and psychology, regression means *"a return to a former or less developed state".*
- The term was introduced by Francis Galton, an English Victorian era polymath, in his 1886 paper *"Regression towards mediocrity in hereditary stature".* In this paper, Galton derived a linear approximation to estimate a son's height from the father's height. He found that a taller-than-average father would have a son that is taller than average, but by a smaller amount than his father is. *Same with shorter-than average fathers.*



Being taller and shorter than average can be "accidental", and modern-day regression is used to find the most generally applicable mapping rule **despite** the presence of accidental and imperfect data.

- Regression a type of **supervised learning**;
- When the term regression is used, the input and output are more often **continuous values**. However, we will show later that regression can be adapted to do classification where the output is discrete (categorical).



Linear Regression

I. Terminology (link to a nice post)

- Variables (known):
  - Independent/explanatory variable (left-hand-side)
  - Dependent/response/target variable (right-hand-side)
- Coefficient and bias (to be learned):
  - Coefficients: multipliers to the independent variables (slope in one dimension)
  - Biases: additive terms to the left-hand-side of the equation (intercept in one dimension)
- Number of variables:
  - Univariate (single independent, single dependent): $ax + b = y$, here a is the coefficient and b is the bias
  - Multivariate (multiple independent, single dependent): $a_1 x_1 + \cdots + a_n x_n + b = y$
  - Multi-response (multiple dependent):

Linear Regression

I. Univariate and evaluation metrics (regression_1_linear.ipynb, Section 1.1)

Formula: $ax + b = y$

Data: A sequence of input-output pairs $(x_1, y_1), \ldots, (x_n, y_n)$.

Goal: Find a and b such at the mean squared error (MSE) is minimized.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - ax_i - b)^2$$

Loss function: $L = \sum_{i=1}^{n}(y_i - ax_i - b)^2$

Analytic solution: Find $\hat{a}$ and $\hat{b}$, such that $\frac{\partial L}{\partial a} = \frac{\partial L}{\partial b} = 0$.

$$\hat{a} = \frac{\sum x_i \sum y_i - n\sum x_i y_i}{(\sum x_i)^2 - n\sum x_i^2}$$

$$\hat{b} = \frac{\sum x_i \sum x_i y_i - \sum x_i^2 \sum y_i}{(\sum x_i)^2 - n\sum x_i^2}$$

How well does a regression work? There are a few quantities that measure how well the regression approximate the real data points. Let $\bar{y} = \frac{1}{n}\sum y_i$, $\bar{x} = \frac{1}{n}\sum x_i$, $\hat{y}_i = \hat{a}x_i + \hat{b}$.

R-squared value:

$$R^2 = 1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \bar{y})^2}$$

P value: Let the null hypothesis be "the independent variable has no explanatory power to the dependent variable", P value is the probability of rejecting the null hypothesis *mistakenly*. The smaller P value is, the more likely that the independent variable explains the dependent variable.

Standard error:

$$\sqrt{\frac{\sum(y_i - \hat{y}_i)^2}{(n-2)\sum(x_i - \bar{x})^2}}$$

Linear Regression

II. Multivariate (regression_1_linear.ipynb, Section 1.2)

Formula: $\langle \vec{a},\ \vec{x} \rangle + b = y$, here $\langle \cdot,\ \cdot \rangle$ means inner product, and $\vec{x}$ is a vector of dimension p.

Data: A sequence of input-output pairs $(\vec{x}_1, y_1), \ldots, (\vec{x}_n, y_n)$.

Loss function: $L = \sum_{i=1}^{n} (y_i\ - \langle \vec{a},\ \vec{x}_i \rangle - b)^2$

Goal: Find vector $\vec{a}$ and scale b such at the L is minimized.

Analytic Solution: Let

Design matrix $\longrightarrow$ $$X = \begin{bmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1p} \\ 1 & x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_{n1} & x_{n2} & \cdots & x_{np} \end{bmatrix}, \qquad \vec{y} = (y_1, y_2, \cdots, y_n)^T$$

The problem becomes finding a vector $\vec{\alpha} = \begin{pmatrix} b \\ \vec{a} \end{pmatrix}$, such that the error vector $\vec{y}\ - X\vec{\alpha}$ is the shortest, since

$$L = \|\vec{y}\ - X\alpha\|_2^2 (= \vec{y}^T\vec{y} - 2\alpha^T X^T\vec{y} + \vec{\alpha}^T X^T X\vec{\alpha})$$

And the analytic solution is $\hat{\alpha}$ such that $\frac{\partial L}{\partial \alpha} = 0$. (here is a link to how to take derivative with vector and matrix)

Finally, we have the analytic solution to multilinear regression problem

$$\hat{\alpha} = (X^T X)^{-1} X^T \vec{y}$$

Non-linear Regression

I. What happens when linearity is not enough

Copyright: MIKKI RAIN/SCIENCE PHOTO LIBRARY

What if Newton insisted on finding a linear relation between the mass of the two objects and distance between them, while the true relation being:

$$\text{Gravitational force} \propto \frac{m_1 * m_2}{r^2}$$

Well, Newton would be as grumpy and depressed as he could possible be.

Featurization:

- **Data:** the raw input
- **Features:** quantities we can cook up with the inputs
- **Example:** To help Sir Newton's with his plight, we just need to take logarithm of both sides of the equation and get:

$$\log(\text{gravitational force}) = \log m_1 + \log m_2 - 2\log r$$

Note that the right-hand side is again a linear combination.

Here, masses m_1, m_2, and distance r are data, but $\log m_1$, $\log m_2$, and r are features.

Non-linear Regression

II. Repurpose regression for categorical prediction (regression_2_logistic.ipynb)

Why logistic regression:

There are many important research topics for which the dependent variable is discrete or categorical. For example: vote for or against, morbidity, or mortality. Logistic regression is a type of regression analysis designed to handle discrete dependent variable.

Logit:

Let Y be a binary random variable with the probability of $Y = 1$ equals p, the **log odds** (or **logit**) of the event $Y = 1$ equals $\log\frac{p}{1-p}$.

Sigmoid function $\sigma(x) = \frac{1}{1+e^{-x}}$:

Sigmoid function is the inverse of the logit function, that is, if $x = \log\frac{y}{1-y}$, then $y = \frac{1}{1+e^{-x}}$.

A statistical view to linear regression:

Let Y be a random variable whose conditional mean on a bunch of explanatory variables $x_1, \dots, x_n$ follows a linear relation of the variables. That is

$$Y = \langle \vec{a}, \vec{x} \rangle + b + \varepsilon$$

where ε is a random variable with zero mean. The process of solving a linear regression is to find the exact relation given **samples** of the random variable Y.

A statistical view to logistic regression:

Let Y be a binary random variable with parameter $p = P(Y = 1)$, we assume that the logit (instead of mean) follows a linear relation with the independent variables $x_1, \dots, x_n$. Mathematically, we write it as

$$\langle \vec{a}, \vec{x} \rangle + b = \log\frac{p}{1-p}$$

Non-linear Regression

II. Repurpose regression for categorical prediction (regression_2_logistic.ipynb), cont.

Motivation to the cross-entropy loss for logistic regression:

Let Y be a binary random variable with parameter $p = P(Y = 1)$. Then we have the cross-entropy function

$$CE(q) = -p \log q - (1-p)\log(1-q)$$

minimized at $q = p$. This implies that, with large enough sample $y_1, \dots, y_n$ of Y, the following function

$$\sum -y_i \log q - (1-y_i)\log(1-q)$$

should also minimized at $q = p$ since $\frac{1}{n}\sum y_i$ converges to p while $\frac{1}{n}\sum(1 - y_i)$, $1-p$.

Final formulation of logistic regression:

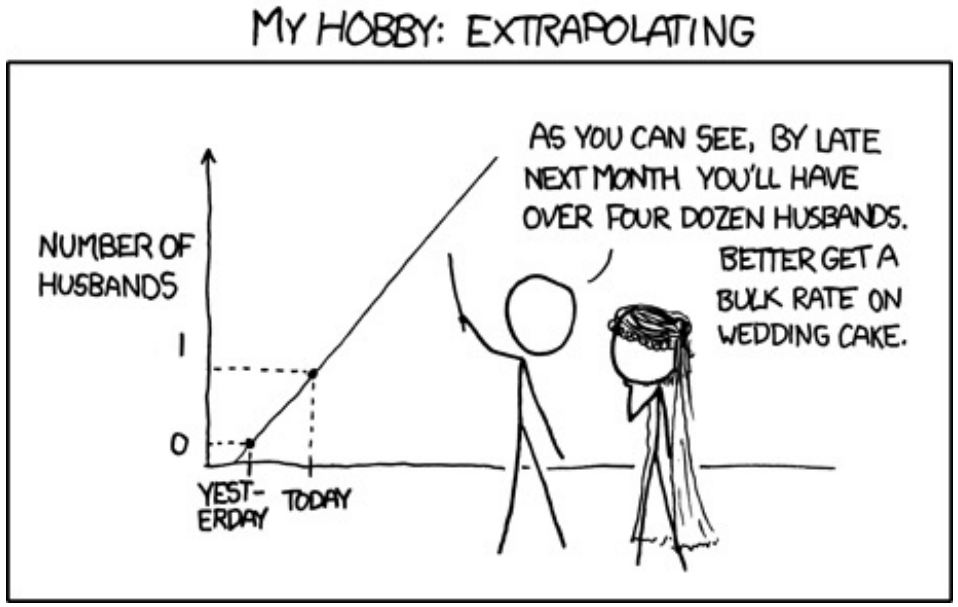
Data: A sequence of input-output pairs $(\vec{x}_1, y_1), \dots, (\vec{x}_n, y_n)$, where y_i is either 0 or 1.

Goal: find vector $\vec{a}$ and scalar b such that

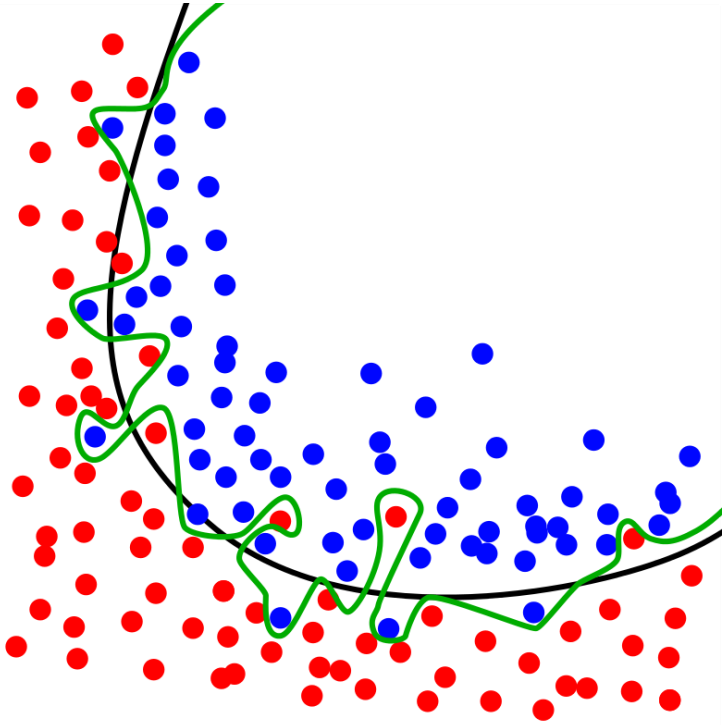
$$L_{CE} = \sum_{i=1}^{n} -y_i \sigma(\langle \vec{a}, \vec{x} \rangle + b) - (1-y_i)\sigma(-\langle \vec{a}, \vec{x} \rangle - b)$$

is minimized. Note that $1 - \sigma(x) = \sigma(-x)$ and hence the second term in L_{CE}.

Regression goes wrong, regularization saves the day



https://xkcd.com/605/



Wiki https://en.wikipedia.org/wiki/Overfitting

Regression goes wrong, regularization saves the day

(regression_3_regularization.ipynb)

Some reasons for regression going wrong:

1. Unjustified extrapolation;
2. Excessively complicated model that overfits on the seen (train) data and fails to generalize to unseen (test) data.

What is regularization:

Regularization is the process of adding information in order to solve an ill-posed problem or to prevent overfitting.

In regression, the added information usually reads like: "The linear coefficients are small or sparse". And the goal is achieved by penalized big coefficients in the loss function.

Commonly used regularization approaches for regression:

Let $\vec{\alpha} = (\vec{a}, b)^T$

- Lasso (sparse features):

$$L = \sum_{i=1}^{n} (y_i - \langle \vec{a}, \vec{x}_i \rangle - b)^2 + \lambda(\|\vec{\alpha}\|_1)$$

- Ridge (reduce the impact of unimportant feature):

$$L = \sum_{i=1}^{n} (y_i - \langle \vec{a}, \vec{x}_i \rangle - b)^2 + \lambda(\|\vec{\alpha}\|_2^2)$$

- ElasticNet (a little of both):

$$L = \sum_{i=1}^{n} (y_i - \langle \vec{a}, \vec{x}_i \rangle - b)^2 + \lambda(\|\vec{\alpha}\|_1 + \|\vec{\alpha}\|_2^2)$$

Analytic solution may no longer exists. Iterative method (e.g. gradient descend) may be used to train the model.

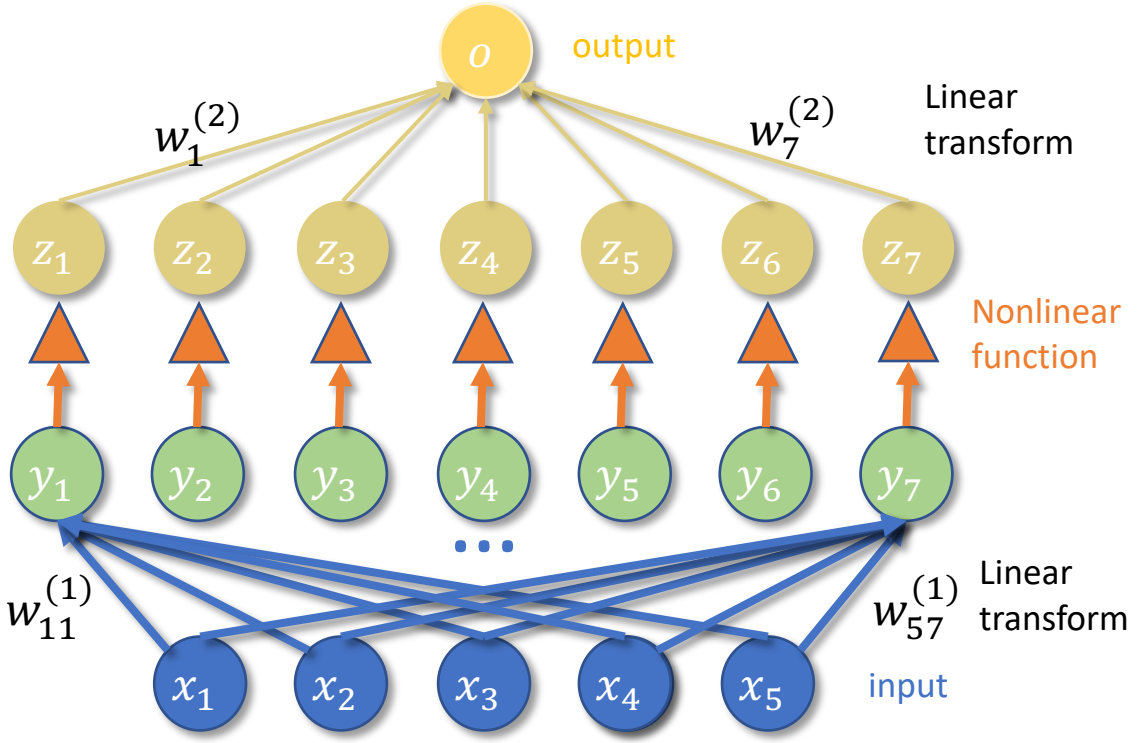
From Regression to Neural Network

(regression_4_fromRegressionToNeuralNetwork.ipynb)

How?

Well, we stack them, and sandwich some non-linearity in between.

We know that the composition of linear maps is still a linear map, so we won't get anything new just by doing linear regression twice, or more. However, if we pass the output from the first regression through some nonlinear function (for example, the sigmoid function in the logistic regression), and then pass the result onto a second regression, we might be able to get something new!



From Regression to Neural Network

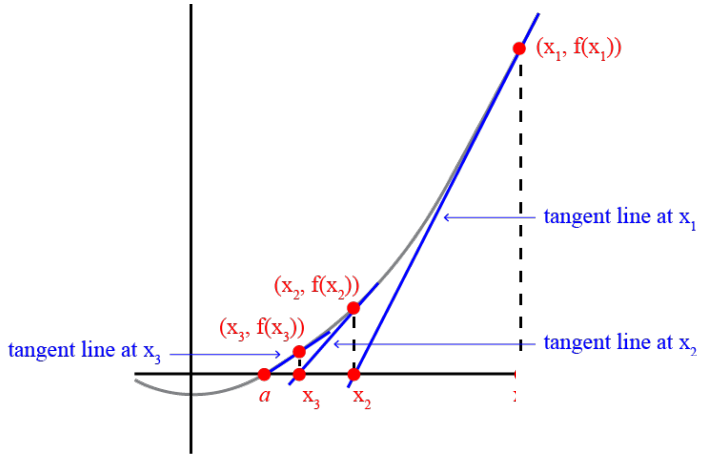
Two approaches to hard-to-solve problem:

1. Simple or no featurization but deep neural network;
   - Few domain knowledge is needed;
   - Tend to lack interpretability.
2. Complex featurization but shallower neural network.
   - Require reliable domain knowledge;
   - Easier to interpret.

Analytic solution v.s. iterative method:

When the model gets complicated, analytic solution may not longer exists and iterative method are needed to train the model. Here are some examples of iterative method:

- Newton's method;
- Gradient descent;
- Stochastic gradient decent;



More on Newton's method here

More on gradient descent here