Regression

October 21, 2021

DSS monday.com consultant meeting

By Pooja Eega and Kevin Miao

Background Picture: NVIDIA GauGAN output by Paragism

Objective

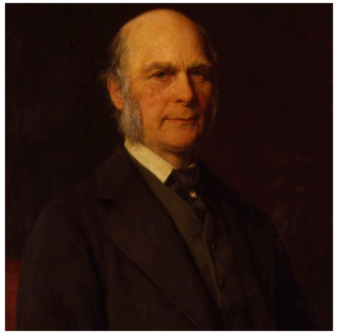
- High level overview of regression and related ML concepts. *Slides*
- Transforming these perspectives into code. *ipynb Demo*

For some, this will be a review; for some, this might be entirely novel content. Those who have questions about the slides are encouraged to address these to their teammates.

History | 1

- Statistical 'Regression' was discovered in the early 19th century
- Often discovery has been ascribed to Gauss, Legendre and Sir Francis Galton
- Sir Francis Galton coined the term 'Regression'
- *Regression towards the mean*: The observation that the descendants of tall ancestors, often were shorter than their parents.

    - Interesting: Galton worked on multiple eugenics projects.



Objective

- **Context:** We have numerical data:
  - $(\vec{x}_i, y_i)$ *where* $x \in \mathbb{R}^d, y \in \mathbb{R}\ \forall\, i \in \{1, \dots, n\}$
  - In other words, we have features, independent variables decoded as x. The predictor is encoded as a y. Each datapoint is such pair.
  - Example:
    - x: square footage, neighborhood, distance to nearest station, number of bedrooms/bathrooms, etc.
    - y: rent
- **Goal:** **Predict** a **numerical outcome** y by reconstructing the underlying true function.
- **Assumption:** There exists some function that models the true relationship between x and y.

Objective

More Concrete Goal:

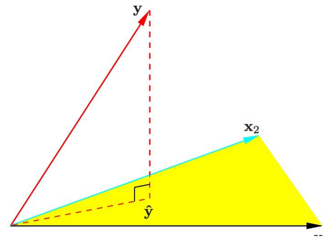
We want to find the $\hat{a}$ **as close as possible** to a in the true function:

$$y = a_1 x_1 + \ldots + a_d x_d$$

- Data 8 perspective (2D):
  - The correlation coefficient ρ turns out to be the slope of the regression line of the variables in standard units.
  - Convert this correlation coefficient ρ to an alpha through multiplying by the ratio of the SD of y and x.
  - Calculate the intercept by plugging in the mean of y and x.
- Minimizing Loss Function Perspective:
  - We can also just try to minimize our loss function, MSE, **least squares.**
  - Through *Stochastic Gradient Descent*, we can attain the particular weights that minimize the error.

OLS/MLE

- Linear Algebra Perspective:
  - It turns out that the least square solution by (b):



$$\hat{a} = \underset{a}{argmax} \|Xa - y\|^2$$

This function that minimizes the distance of each residual to the function can also be interpreted as a projection of each point onto a smaller hyperplane.

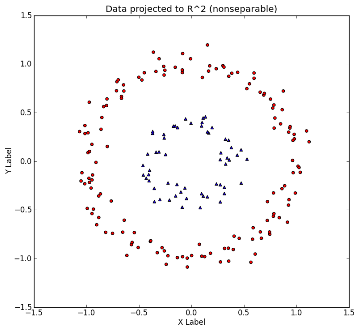
- MLE Perspective:
  - It also turns out that the $OLS\ solution$ we find, is the most likely to occur given the data that we observe.

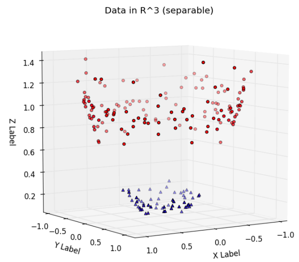
$$\hat{a} = \underset{a}{argmax} P(y\,|X, a)$$

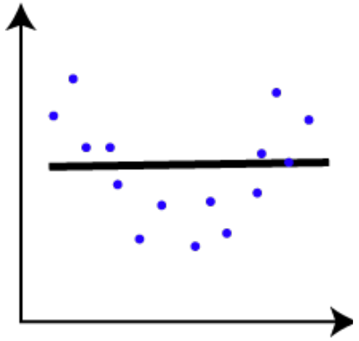
- If you want to know more about the Statistics perspective, ask Ewen!

Lifting/Kernel Trick | 5

- However, what happens when we wanted to predict a nonlinear function?

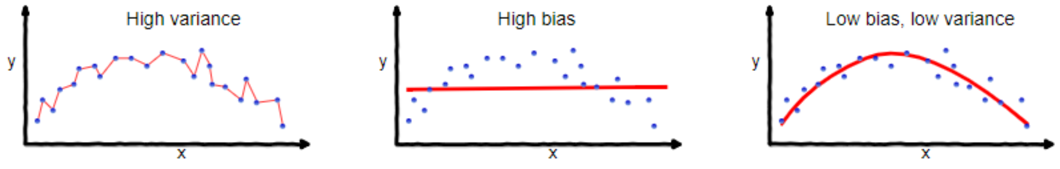






- We can combine turn our feature x into features x, x^2 , called lifting.
- The Kernel trick bypasses the direct need to create this featurization but does put a prior belief on our features, which has consequences.

- **Loss Function:** $MSE = \frac{1}{n}\sum_{i=1}^{n}(pred - actual)^2 = \frac{1}{n}||X\hat{w} - y||^2$
  - **Can be decomposed into** bias² **and** variance
    - Bias: How well does my model fit the data without noise.
    - Variance: How much noise am I taking into account.

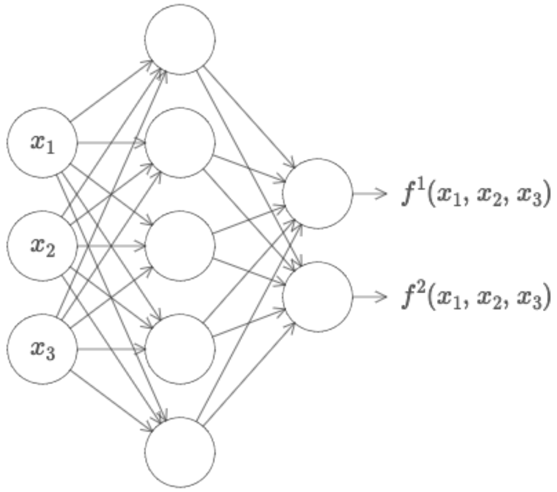


overfitting underfitting Good balance

- To counteract *high variance/too much noise*, we **regularize;** add an extra penalty term to the loss function so that weights don't explode:

RIDGE (Complex Models): $\lambda||w||_2^2$	LASSO (Parsimony): $\lambda\,||w||_1$

- **Nonlinearities** are a bottleneck!
- Artificial Neural Networks provide solution:
  - **Universal Approximation Theorem** states that *neural networks* can approximate any continuous function.
  - **Moreover,** any neural network with one hidden layer can approximate any real function up to your chosen precision.
- Caveats: High dimensionality → Overfitting



- Recall:
  - Traditionally, the MSE can be decomposed as:
    - **Bias^2 :** How well would my model fit without any noise?
    - **Variance**: How much contamination do I take into my model?
  - What happens if we have more way more variables in our model than there are in real life for a certain pattern? In other words, what happen if we overparameterize?

Bias, Variance, and Regularization

- Recall:
  - Traditionally, the MSE can be decomposed as:
    - **Bias^2 :** How well would my model fit without any noise?
    - **Variance**: How much contamination do I take into my model?
  - What happens if we have more way more variables in our model than there are in real life for a certain pattern? In other words, what happen if we overparameterize?
  - **<u>Summer '19</u>:**

Reconciling modern machine-learning practice and the classical bias–variance trade-off

Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal

PNAS August 6, 2019 116 (32) 15849-15854; first published July 24, 2019; https://doi.org/10.1073/pnas.1903070116

Edited by Peter J. Bickel, University of California, Berkeley, CA, and approved July 2, 2019 (received for review February 21, 2019)

Demo

End of slides