Instituto de Computação
UNIVERSIDADE ESTADUAL DE CAMPINAS

Capacitação profissional em tecnologias de Inteligência Artificial

Machine Learning Overview

Prof. Edson Borin
https://www.ic.unicamp.br/~edson
Institute of Computing - UNICAMP

Common ML Algorithms

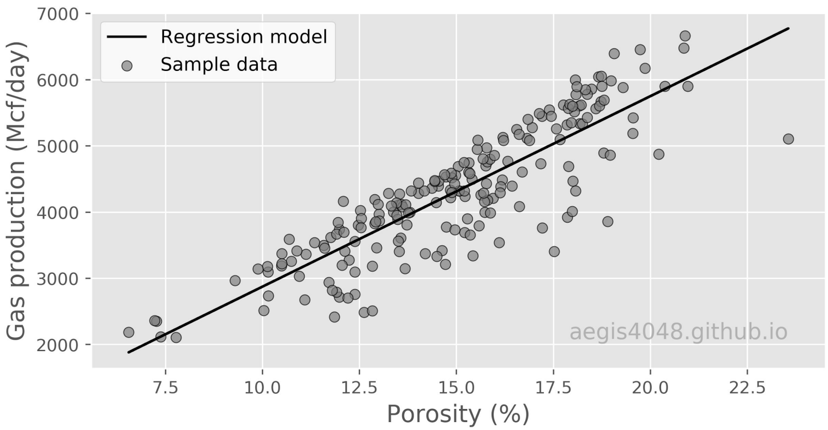
- **Machine learning**
  - **Supervised Learning**
    - Classification
      - Logistic regression
      - SVM
      - Neural Network
      - Decision Tree
      - Random Forest
      - GBDT
      - KNN
      - Naive Bayes
    - Regression
      - Linear regression
      - SVM
      - Neural Network
      - Decision Tree
      - Random Forest
      - GBDT
  - **Unsupervised Learning**
    - Clustering
      - K-means
      - Hierarchical clustering
      - Density-based clustering
    - Others
      - Correlation rule
      - Principal component analysis (PCA)
      - Gaussian mixture models (GMM)

Linear Regression

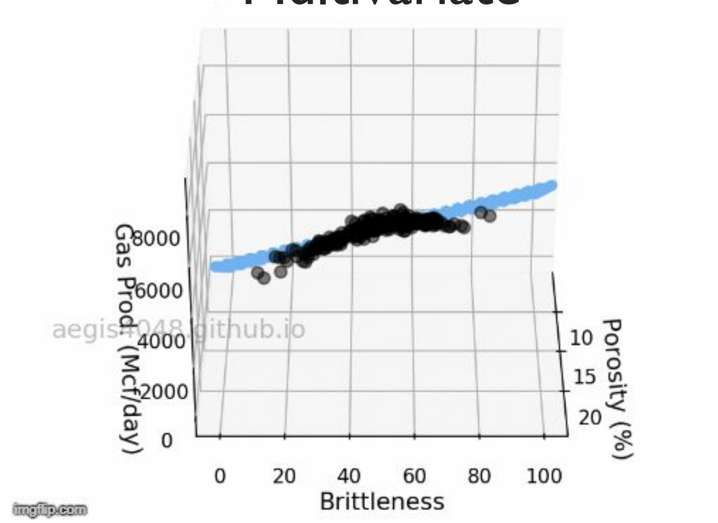
Linear regression

- Statistical analysis method to determine linear relationships between a scalar response (output variable or label) and one or more explanatory variables (input variables or features)

Univariate



Multivariate



Source: https://aegis4048.github.io/mutiple_linear_regression_and_visualization_in_python

Linear regression

Let:

- θ: Set of model parameters. $\theta = \{\theta_0, \theta_1, \ldots, \theta_n\}$
- x: Set of input variables. $x = \{x_1, x_2, \ldots, x_n\}$
- $\hat{y}$: model output (predicted value)
- $\hat{y} = \theta_0 + \theta_1 \cdot x_1 + \ldots + \theta_n \cdot x_n$
- $\hat{y} = \theta_0 + \theta_1 \cdot x_1$
- $h_\theta(x) = \theta_0 + \theta_1 \cdot x_1 + \ldots + \theta_n \cdot x_n$

Multivariate linear model

Univariate linear model

Also known has hypothesis!

Linear regression

Linear model

- $h_\theta(x) = \theta_0 + \theta_1 \cdot x_1 + \ldots + \theta_n \cdot x_n$

Linear regression

Linear model

- $h_{\theta}(x) = \theta_0 + \theta_1 \cdot x_1 + \ldots + \theta_n \cdot x_n$

Also explained as:

- $h_w(x) = \mathbf{b} + \boldsymbol{w_1} \cdot x_1 + \ldots + \boldsymbol{w_n} \cdot x_n$
- $h(x) = \mathbf{intercept} + \boldsymbol{coef_1} \cdot x_1 + \ldots + \boldsymbol{coef_n} \cdot x_n$

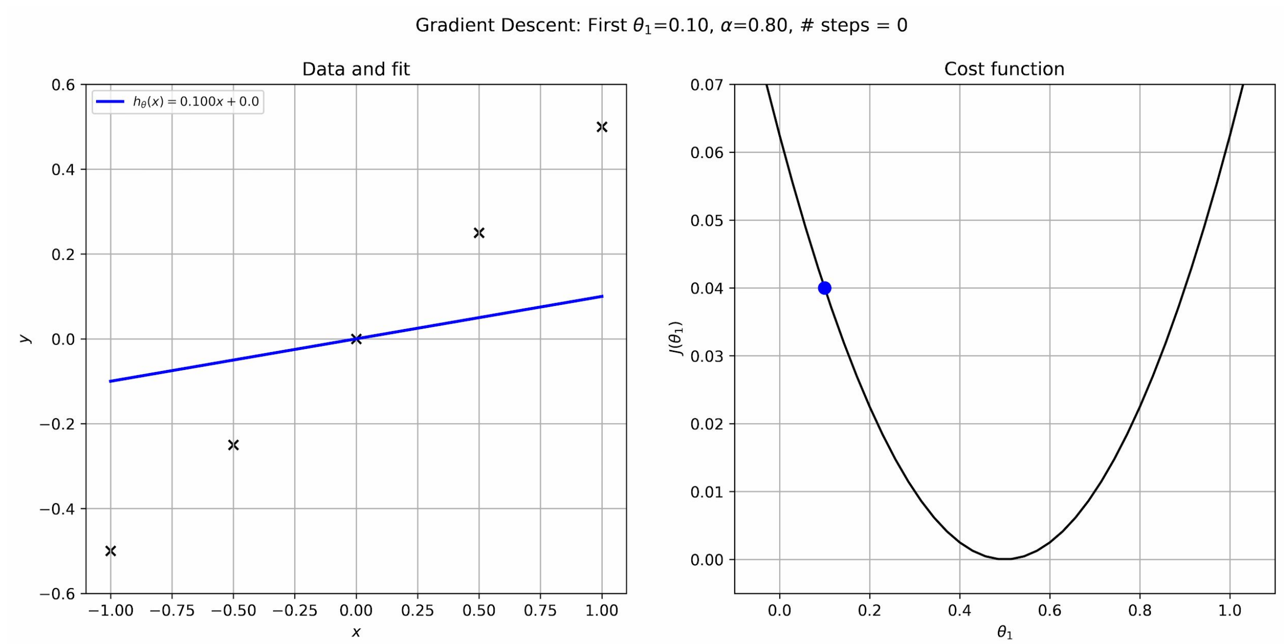
Linear regression

Linear model

- $h_\theta(x) = \theta_0 + \theta_1 \cdot x_1 + \ldots + \theta_n \cdot x_n$

Training

Select θ so that the cost function is minimized



Linear regression

Linear model

- $h_\theta(x) = \theta_0 + \theta_1 \cdot x_1 + \ldots + \theta_n \cdot x_n$

Cost function:

- Usually the Mean Squared Error (MSE)

$$J(\theta) = \underbrace{\frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2}_{\mathrm{MSE}(X, h_\theta)}$$

Linear regression

Can also be the Mean Absolute Error (MAE)

Linear model

- $h_\theta(x) = \theta_0 + \theta_1 \cdot x_1 + \ldots + \theta_n \cdot x_n$

Cost function:

- Usually the Mean Squared Error (MSE)

$$J(\theta) = \underbrace{\frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2}_{\mathrm{MSE}(X, h_\theta)}$$

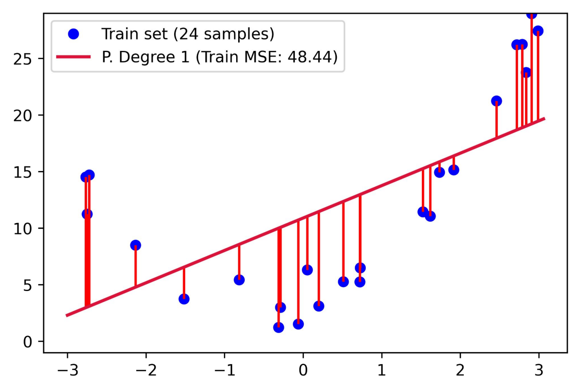
Polynomial regression

Similar to linear regression.

<u>Polynomial model</u>

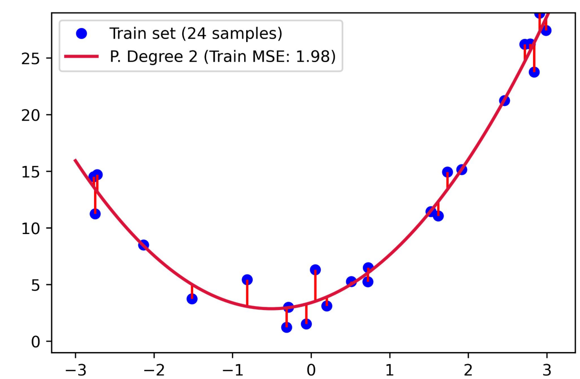
- $h_\theta(x) = \theta_0 + \theta_1 \cdot x_1 + \theta_2 \cdot x_1^{\,2} + \theta_3 \cdot x_1^{\,3} + \ldots + \theta_n \cdot x_1^{\,n}$

Linear model



$f(x) = 10.895022 + 2.863998\, x$

Poly. degree = 2



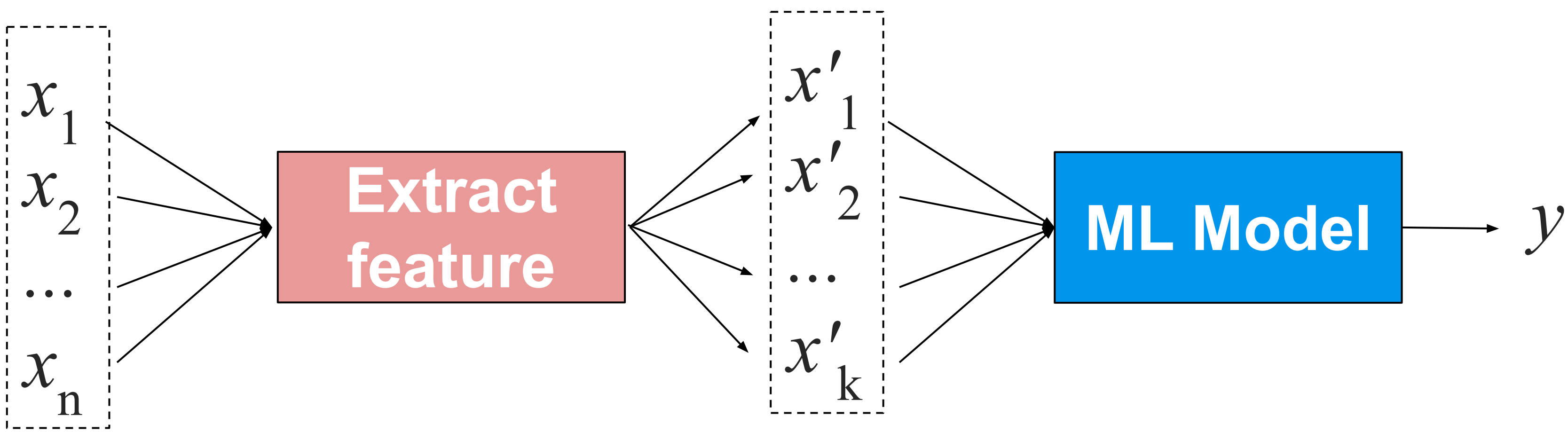
$f(x) = 3.403875 + 2.122188\, x + 2.099980\, x^2$

Polynomial regression

Similar to linear regression.

<u>Polynomial model</u>

- $h_\theta(x) = \theta_0 + \theta_1 \cdot x_1 + \theta_2 \cdot x_1^{\,2} + \theta_3 \cdot x_1^{\,3} + \ldots + \theta_n \cdot x_1^{\,n}$

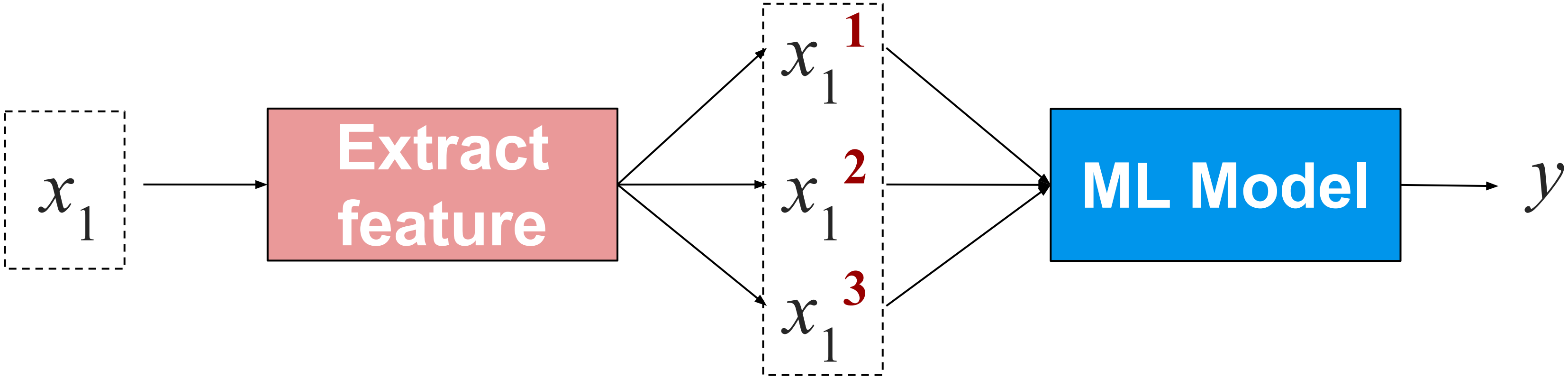


Polynomial regression

Similar to linear regression.

<u>Polynomial model</u>

- $h_\theta(x) = \theta_0 + \theta_1 \cdot x_1 + \theta_2 \cdot x_1^2 + \theta_3 \cdot x_1^3 + \ldots + \theta_n \cdot x_1^n$

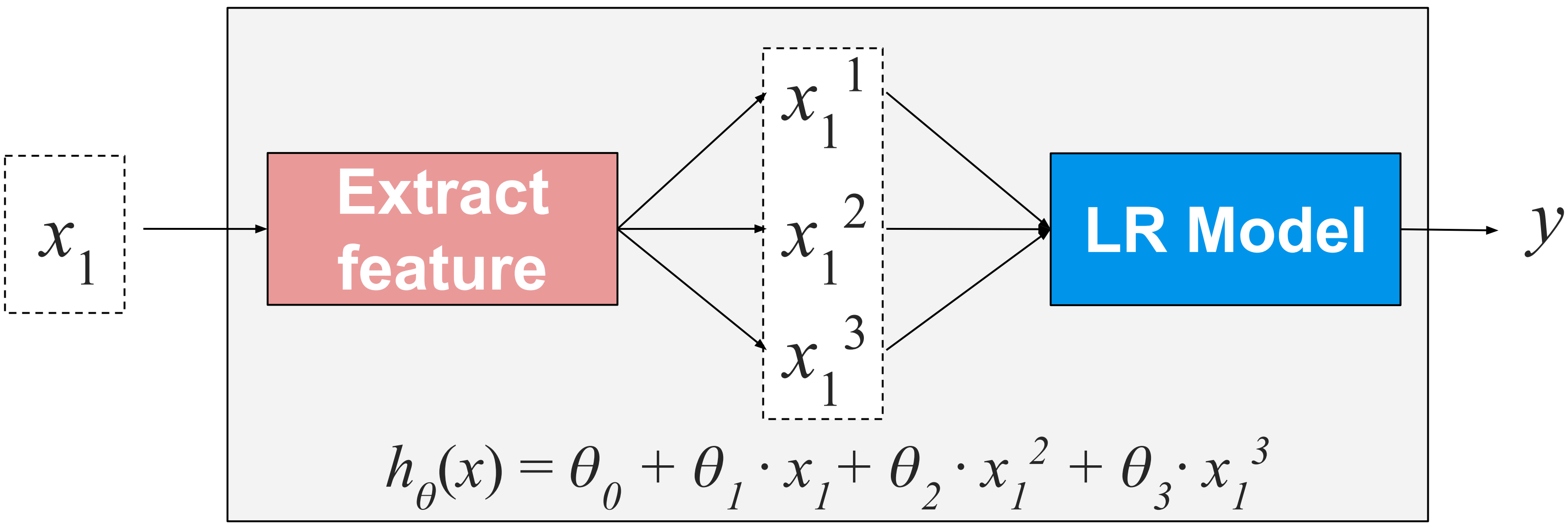


Polynomial regression

Similar to linear regression.

<u>Polynomial model</u>

- $h_\theta(x) = \theta_0 + \theta_1 \cdot x_1 + \theta_2 \cdot x_1^{2} + \theta_3 \cdot x_1^{3} + \ldots + \theta_n \cdot x_1^{n}$



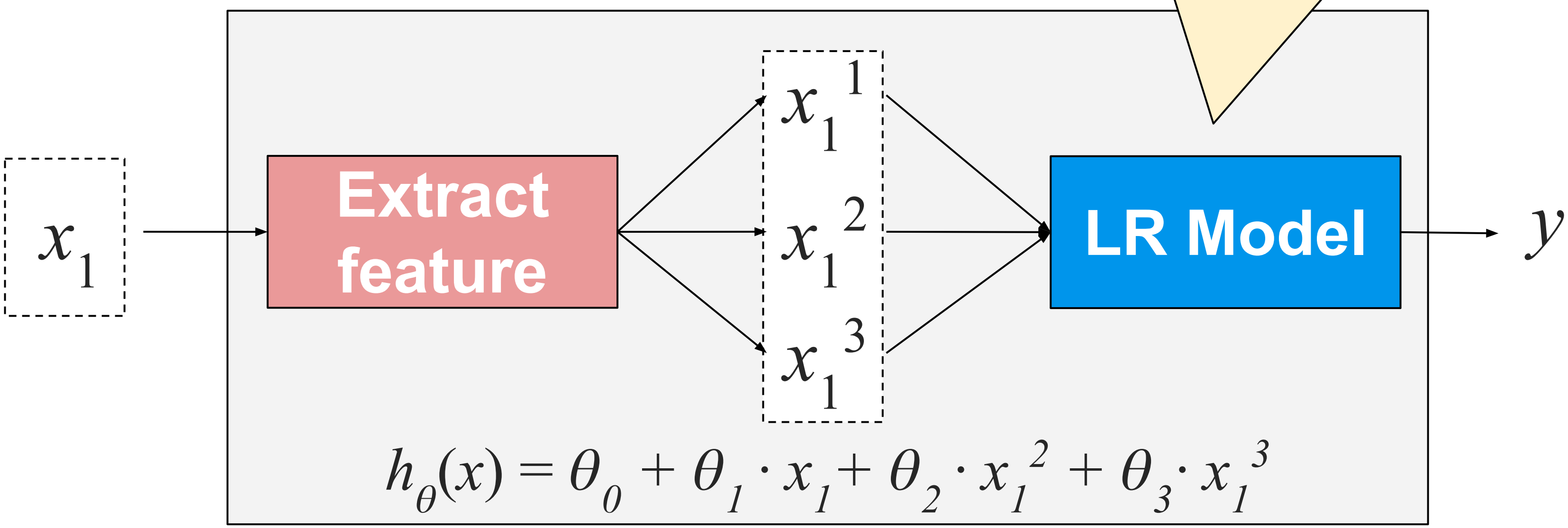
Polynomial regression

Similar to linear reg

Polynomial model

- $h_\theta(x) = \theta_0 + \theta_1 \cdot x_1 + \theta_2 \cdot x_1^2 + \theta_3 \cdot x_1^3 + \ldots + \theta_n$

Train in the same way we train linear regression models!



Supervised Learning Algorithms
Linear Regression

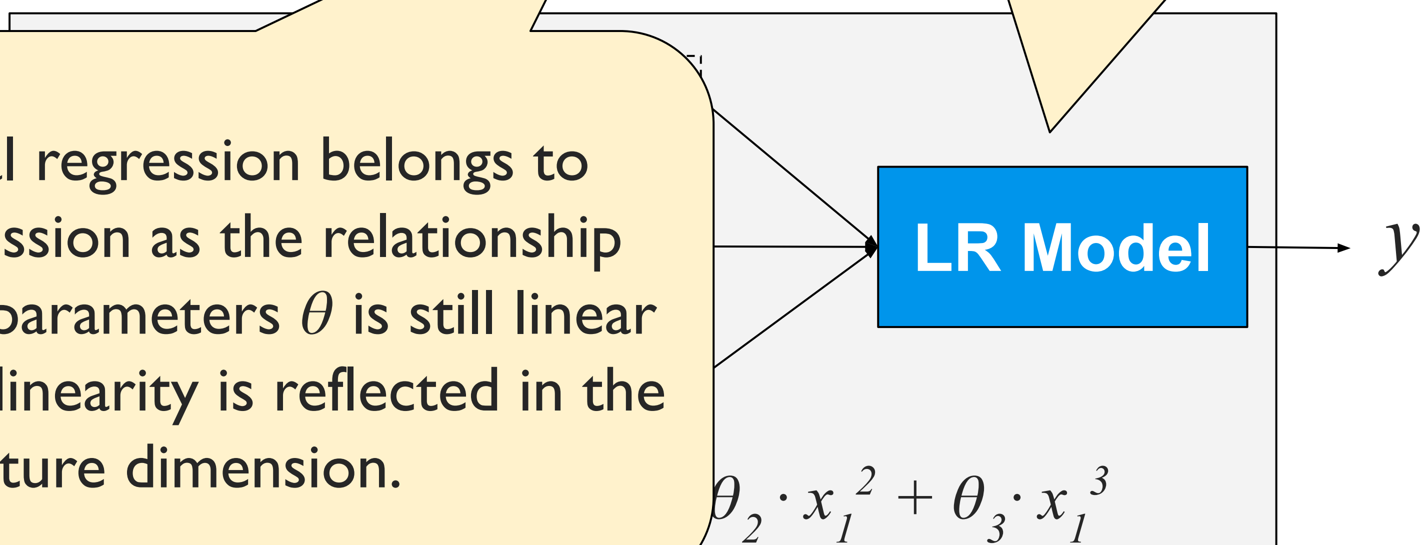
Polynomial regression

Similar to linear reg

<u>Polynomial model</u>

- $h_\theta(x) = \theta_0 + \theta_1 \cdot x_1 + \theta_2 \cdot x_1^2 + \theta_3 \cdot x_1^3 + \ldots + \theta_n$

Train in the same way we train linear regression models!



Polynomial regression belongs to linear regression as the relationship between its parameters θ is still linear while its nonlinearity is reflected in the feature dimension.

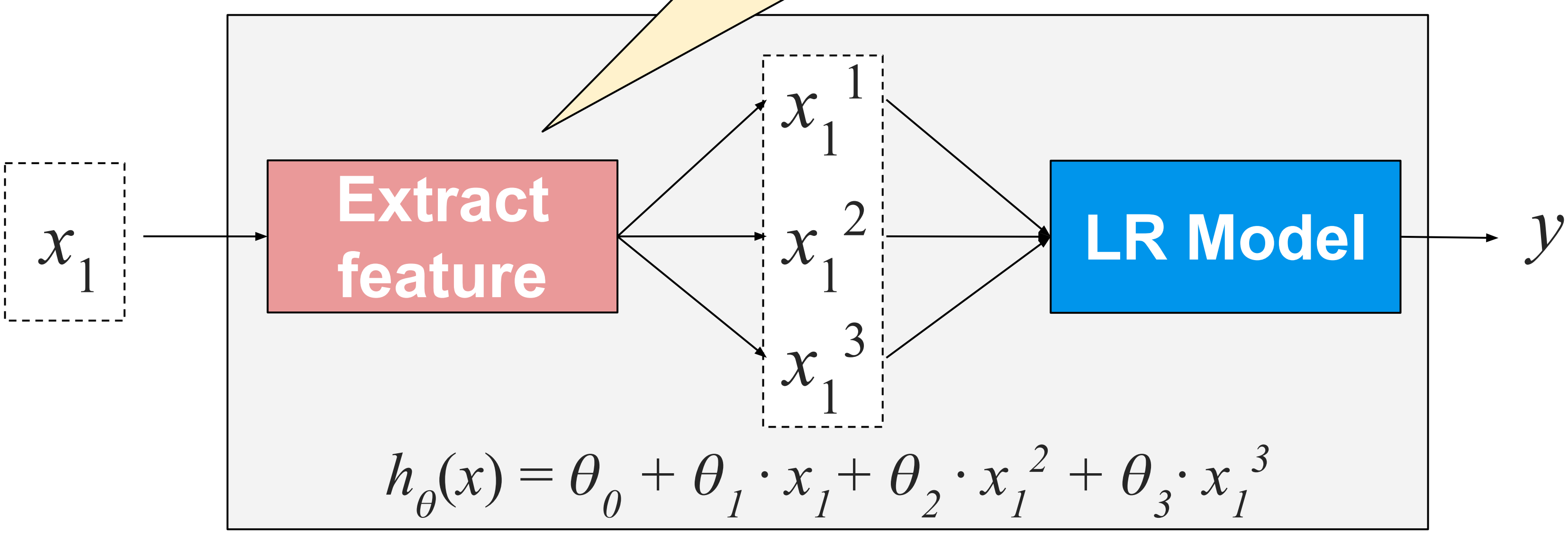
Polynomial regression

Scikit Learn
PolynomialFeatures module

Similar to linear regressi

Polynomial model

- $h_\theta(x) = \theta_0 + \theta_1 \cdot x_1 + \theta_2 \cdot x_1^2 + \ldots + \theta_n \cdot x_1^n$



Ridge/Lasso regression

In linear (and polynomial) regression, the more input variables, the more parameters the model will have, and the greater the chance of overfitting the training data

Ridge/Lasso regression

In linear (and polynomial) regression, the more input variables, the more parameters the model will have, and the greater the chance of overfitting the training data

An approach to minimize overfit is to tweak the training process to produce a model with small parameter (θ) values

Supervised Learning Algorithms
Linear Regression

Ridge/Lasso regression

In linear (and polynomial) regression, the more input variables, the ... have, and the greater the ... e of overfitting the training data

This is called regularization!

An approach to minimize overfit is to tweak the training process to produce a model with small parameter (θ) values

Ridge/Lasso regression

How do we tweak the training process to produce a model with small parameter (θ) values?

Ridge/Lasso regression

How do we tweak the training process to produce a model with small parameter (θ) values?

- By penalizing the cost function with θ values!

$$J(\theta) = \underbrace{\frac{1}{\mathrm{m}} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2}_{\mathrm{MSE}(\mathrm{X}, h_\theta)} + \boxed{(\theta_1^{\ 2} + \theta_2^{\ 2} + \ldots + \theta_n^{\ 2})}$$

Ridge/Lasso regression

How do we tweak the training process to produce a model with small parameter (θ) values?

- By penalizing the cost function with θ values!

θ_0 (a.k.a. bias) is not penalized

$$J(\theta) = \underbrace{\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2}_{\mathrm{MSE}(X,h_\theta)} + \boxed{(\theta_1^{\,2} + \theta_2^{\,2} + \ldots + \theta_n^{\,2})}$$

Ridge/Lasso regression

How do we tweak the training process to produce a model with small parameter (θ) values?

- By penalizing the cost function with θ values!

$$J(\theta) = \underbrace{\frac{1}{\mathrm{m}} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2}_{\mathrm{MSE}(\mathrm{X}, h_\theta)} + \boxed{\alpha \,.\, (\theta_1^{\,2} + \theta_2^{\,2} + \ldots + \theta_n^{\,2})}$$

α: hyperparameter

Ridge/Lasso regression

How do we tweak the training process to produce
a model with small par

- By penalizing the co

Assuming w = $\{\theta_1, \ldots, \theta_n\}$,
then, $(\theta_1^{\,2} + \theta_2^{\,2} + \ldots + \theta_n^{\,2}) = ||\mathrm{w}||_2^{\,2}$

$$J(\theta) = \underbrace{\frac{1}{\mathrm{m}} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2}_{\mathrm{MSE}(\mathrm{X}, h_\theta)} + \boxed{\alpha \,.\, (\theta_1^{\,2} + \theta_2^{\,2} + \ldots + \theta_n^{\,2})}$$

Ridge/Lasso regression

How do we tweak the training process to produce a model with small par…

- By penalizing the co…

Assuming w = $\{\theta_1, \ldots, \theta_n\}$,
then, $(|\theta_1| + |\theta_2| + \ldots + |\theta_n|) = ||w||_1$

$$J(\theta) = \underbrace{\frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2}_{\mathrm{MSE}(X, h_\theta)} + \boxed{\alpha \cdot (|\theta_1| + |\theta_2| + \ldots + |\theta_n|)}$$

Ridge/Lasso regression

$$J(\theta) = \underbrace{\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2}_{\text{MSE}(X,h_\theta)} + \boxed{\alpha \,.\, (\theta_1^{\,2} + \theta_2^{\,2} + \ldots + \theta_n^{\,2})}$$

Ridge regression

$$J(\theta) = \underbrace{\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2}_{\text{MSE}(X,h_\theta)} + \boxed{\alpha \,.\, (|\theta_1| + |\theta_2| + \ldots + |\theta_n|)}$$

Lasso regression

Supervised Learning Algorithms

Line

Ridge/Lasso regre

Tends to produce small parameter values (close to zero), but not zero.

$$J(\theta) = \underbrace{\frac{1}{\mathrm{m}} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2}_{\mathrm{MSE}(\mathrm{X}, h_\theta)} + \boxed{\alpha \;.\; (\theta_1^{\,2} + \theta_2^{\,2} + \ldots + \theta_n^{\,2})}$$

Ridge regression

$$J(\theta) = \frac{1}{\mathrm{m}} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2 + \boxed{\alpha \;.\; (|\theta_1| + |\theta_2| + \ldots + |\theta_n|)}$$

Tends to zero-out parameters associated with non-informative features! Can be used to simplify models!

$$h_\theta(x) = \theta_0 + \theta_1 \cdot x_1 + \ldots + \theta_n \cdot x_n$$

Ridge/Lasso regression

$$J(\theta) = \underbrace{\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2}_{\mathrm{MSE}(\mathrm{X},h_\theta)} + \boxed{\alpha \,.\, (\theta_1^{\,2} + \theta_2^{\,2} + \ldots + \theta_n^{\,2})}$$

Make sure you perform feature scaling if using Ridge/Lasso regression

Elastic Net regression

- Combines both Ridge and Lasso regression
- Hyperparameter r: mix ratio

r: mix ratio

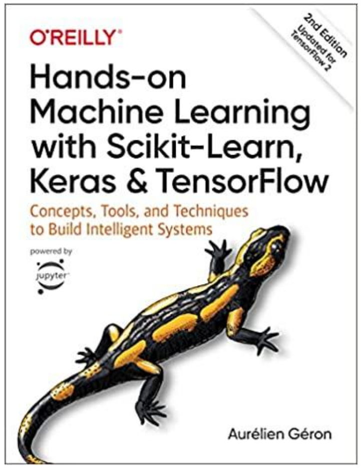
$$J(\theta) = \underbrace{\frac{1}{\mathrm{m}} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2}_{\mathrm{MSE}(\mathrm{X}, h_\theta)} + \underbrace{r.\alpha.\sum_{i=1}^{n} |\theta_i|}_{\text{Lasso Regularization Term}} + (1\text{-}r).\alpha.\underbrace{\sum_{i=1}^{n} (\theta_i)^2}_{\text{Ridge Regularization Term}}$$

General recommendations

- It is usually preferable to have at least a little bit of regularization: Avoid plain LR!
- Make sure you perform feature scaling if using Ridge/Lasso/Elastic Net regression
- Ridge is a good default, but if you suspect only a few features are actually useful, you should prefer Lasso or Elastic Net
  - Both tend to reduce the useless features' weights down to zero.

Supervised Learning Algorithms
Linear Regression

References



- Aurelien Geron. Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems - 2019
  - Chapter 4

- Logistic Regression on the scikit-learn website:
  - https://scikit-learn.org/stable/modules/linear_model.html

Instituto de Computação
UNIVERSIDADE ESTADUAL DE CAMPINAS

Capacitação profissional em tecnologias de Inteligência Artificial

Machine Learning Overview

Prof. Edson Borin
https://www.ic.unicamp.br/~edson
Institute of Computing - UNICAMP

Video Index

Video lessons	Length	Start	Length	Subtopics	
class-4.1: Linear/Polynomial/Ridge/Lasso regressions	0:24:35	0:00:00	0:03:30	Linear Regression	Introduction
		0:03:30	0:02:35		Linear model
		0:06:05	0:02:26		Training / Cost function
		0:08:31	0:05:23	Polynomial Regression	Polynomial regression
		0:13:54	0:06:27	Ridge/Lasso regression	Ridge/Lasso regression
		0:20:21	0:02:13	Elastic Net regression	Elastic Net regression
		0:22:34	0:02:01	General recommendations	General recommendations
		0:24:35			