ECS 171 Machine Learning

Lecture3: Overfitting, underfitting, Polynomial Regression, Regularization, MLE

Instructor: Dr. Setareh Rafatirad

Group Project Reminder

- Roles
  - Data Management
    - Gathering data, data preprocessing, cleaning, metadata extraction, exploratory data analysis
  - Data Visualization
    - EDA, Data understanding
  - Machine learning and Algorithms
    - Executing and developing all aspects of machine learning and developing a machine learning model from feature engineering all the way to building prediction models, and active learning.
  - Quality assurance
    - Evaluation and testing
  - Project management
    - Handling communication between the team members, documentation, deadlines and timely delivery of the project and reports.
    - Use ClickUp (for free) https://clickup.com/pricing

GD Update Rule for 1 sample cont.

- Also called the **Least Mean Squares (LMS) update rule** (or **Widrow-Hoff** learning rule).

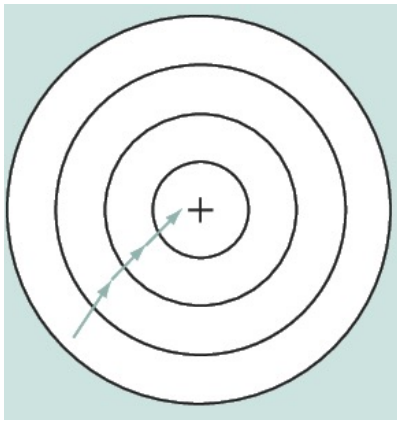
$$w_j = w_j + 2a\,(y^{(i)} - \sum_{k=0}^{n} w_k x_k^{(i)})\; x_j^{(i)}$$

$$w_j = w_j + a\,(y^{(i)} - w^T x^{(i)}) x_j^{(i)}$$

GD Update Rule for *m* samples

- There are 2 ways to deal with m samples:

Batch gradient descent



Repeat until convergence:

{ for j=1 to n

$$w_j := w_j + a \sum_{i=1}^{m} (y^{(i)} - \boldsymbol{w}^T \boldsymbol{x}^{(i)}) x_j^{(i)}$$

}

Batch represents a sample of data
m: batch size
n: number of attributes

Always converges

Stochastic gradient descent

For 1 epoch

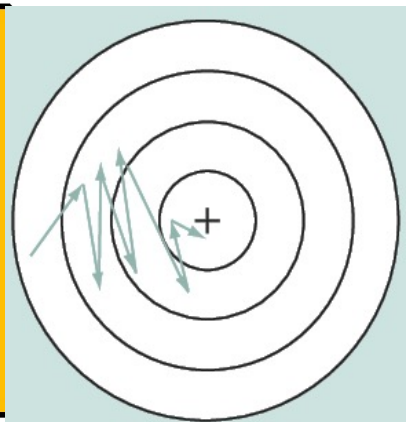
Repeat until convergence:

{ for i= 1 to m
{ for j=1 to n

$$w_j := w_j + a(y^{(i)} - \boldsymbol{w}^T \boldsymbol{x}^{(i)}) x_j^{(i)}$$

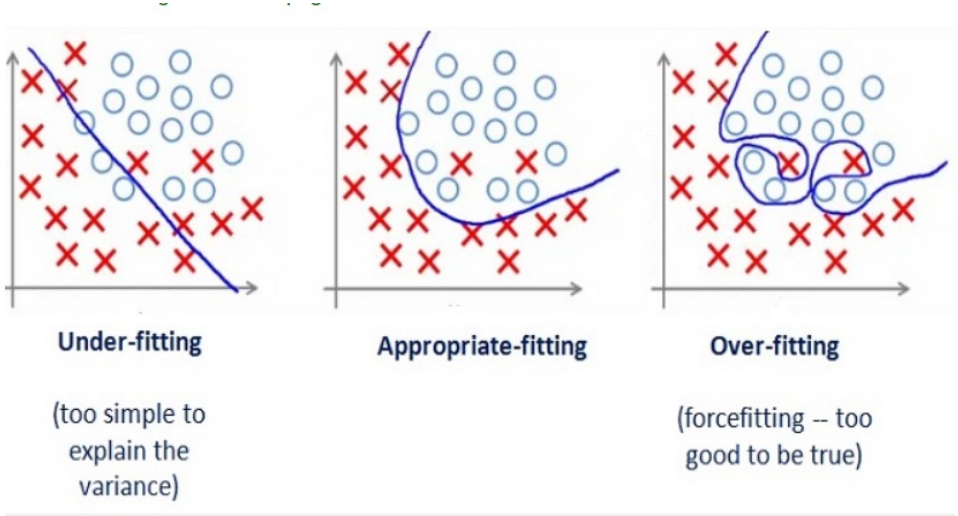
}
}

m: number of observations
n: number of attributes

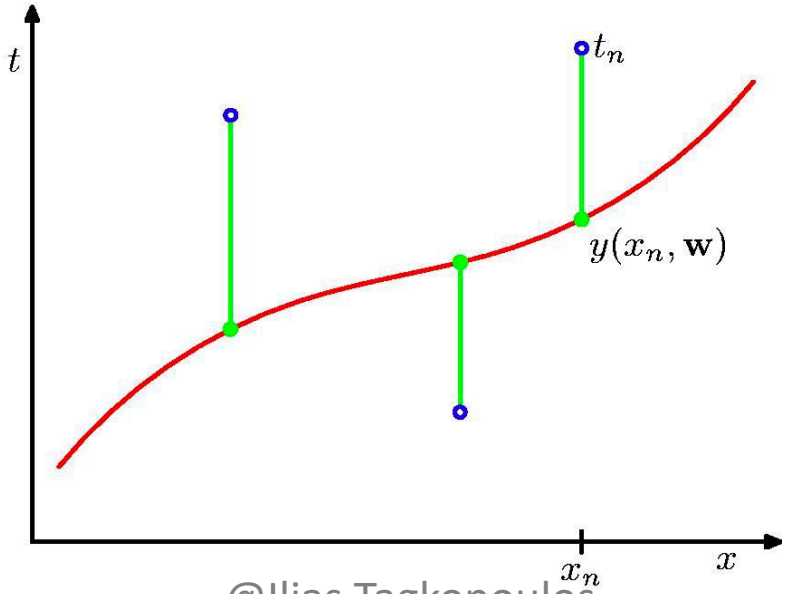


Can take many cycles to converge, or never converge.

Overfitting, Underfitting, Error Function



@Anup Bhande



@Ilias Tagkopoulos

Underfitting occurs when the model fails to learn the relationship between input and output.

Cons: high prediction error.

Model overfitting refers to predicting a very complex model s.t. it fits almost every observation and it learns every variability in the training set.

Cons: The model that overfits is likely to have learned the noise that can obscure the true relationship bw input and output.

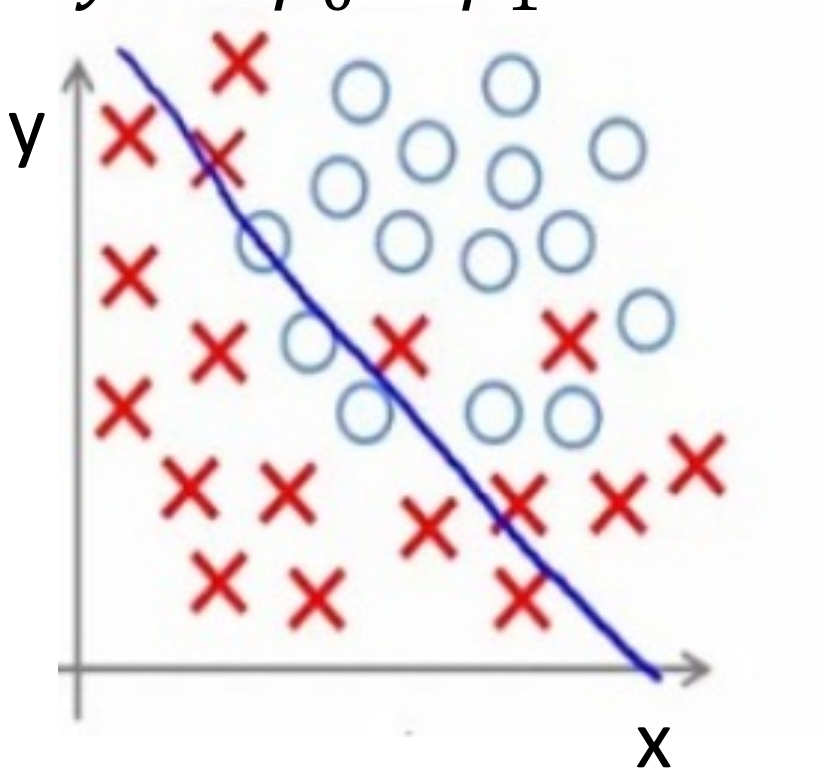
$$y(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \ldots + w_M x^M = \sum_{j=0}^{M} w_j x^j$$

N = number of samples

$$E(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^{N} \{y(x_n, \mathbf{w}) - t_n\}^2$$

Polynomial Regression

$$y = \beta_0 + \beta_1 x$$



Polynomial regression is a form of regression where the relationship between x and y an n^{th} degree polynomial in x.

Polynomial Regression as a Linear Model

- Polynomial function of nth degree polynomial

$$f(x) = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \cdots + \beta_n x^n$$

Dependent values (y)

intercept

Model parameters (weights)

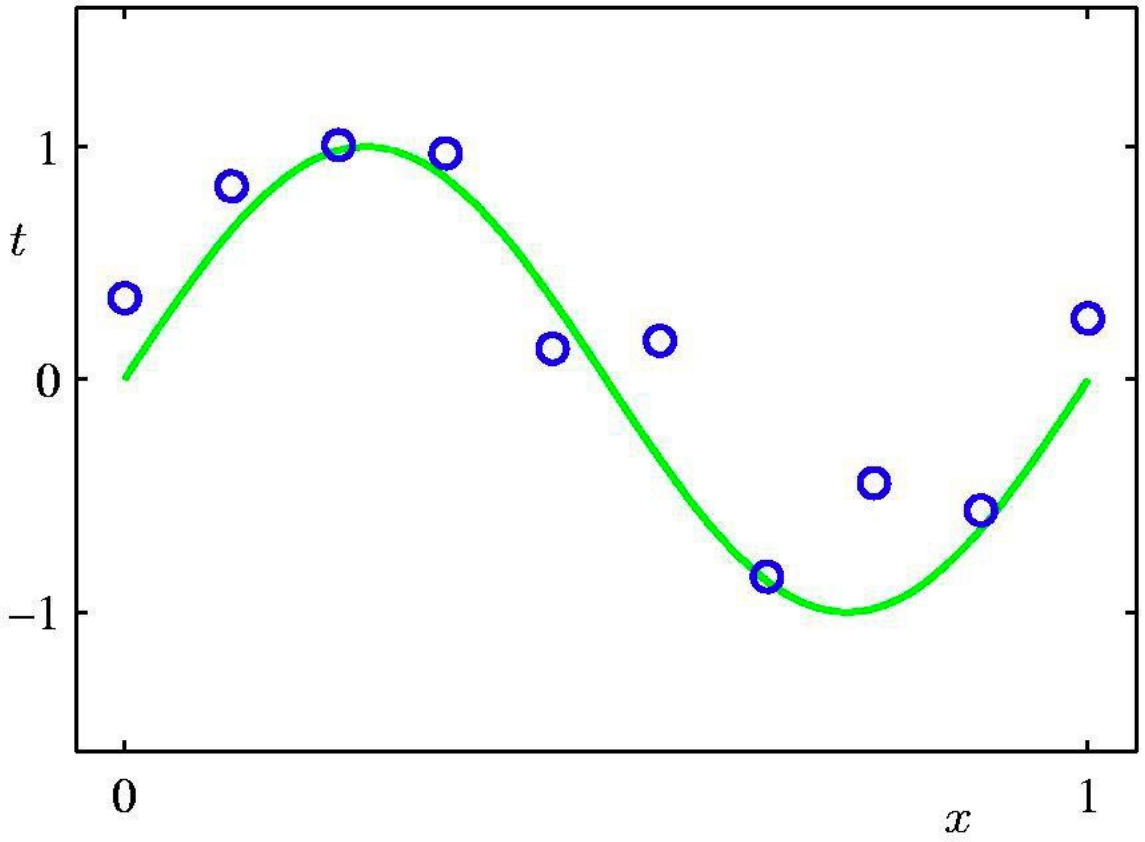
Relabeling

$$f(x) = \omega_0 + \omega_1 x + \omega_2 x^2 + \omega_3 x^3 + \cdots + \omega_n x^n$$

$$f(z) = \omega_0 + \omega_1 z_1 + \omega_2 z_2 + \omega_3 z_3 + \cdots + \omega_n z_n$$

Regression and Sum of Squares Loss Function

- M= number of features
- N= size of a dataset D

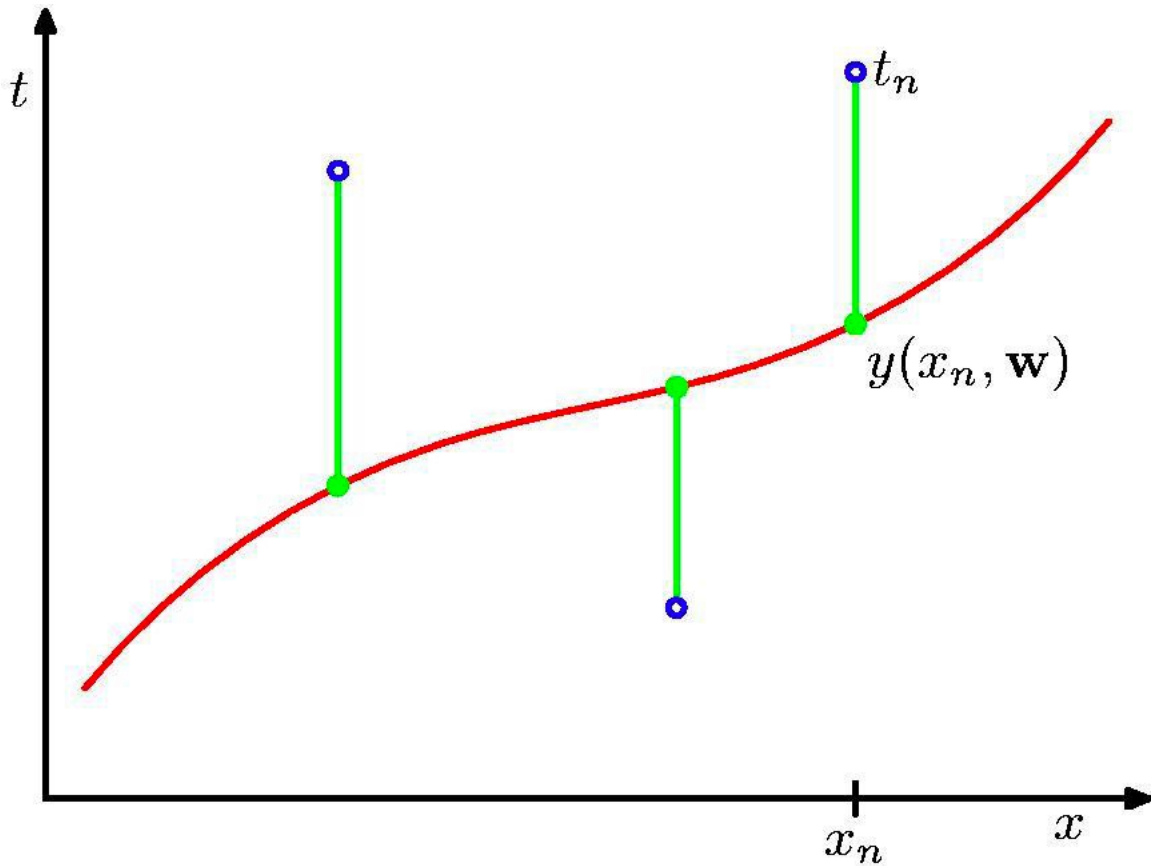


polynomial order M

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

Sum-of-Squares Error Function

- Loss function (cost function, error function)



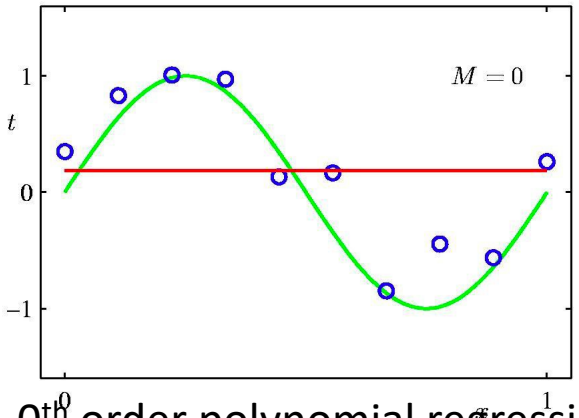
$$E(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^{N} \{y(x_n, \mathbf{w}) - t_n\}^2$$

or

$$E(\mathbf{w}) = \sum_{n=1}^{N} \{y(x_n, \mathbf{w}) - t_n\}^2$$

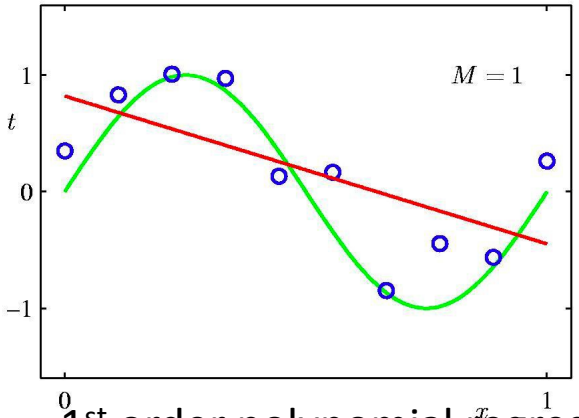
Constructing a Polynomial Regression Model

M= number of features



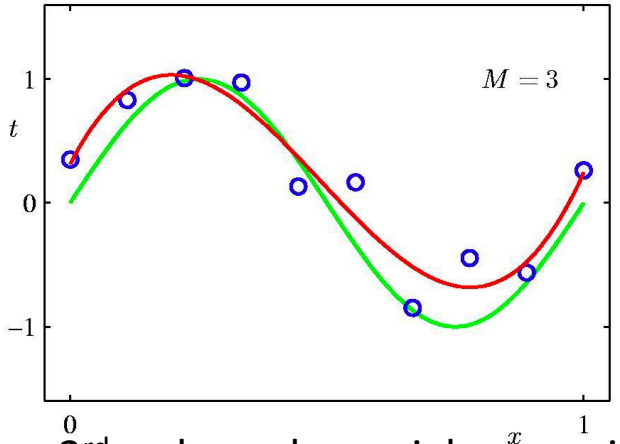
0th order polynomial regression

M=0



1st order polynomial regression

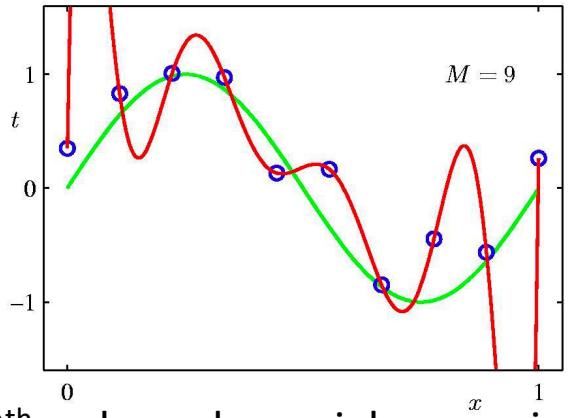
M=1



3rd order polynomial regression

M=3

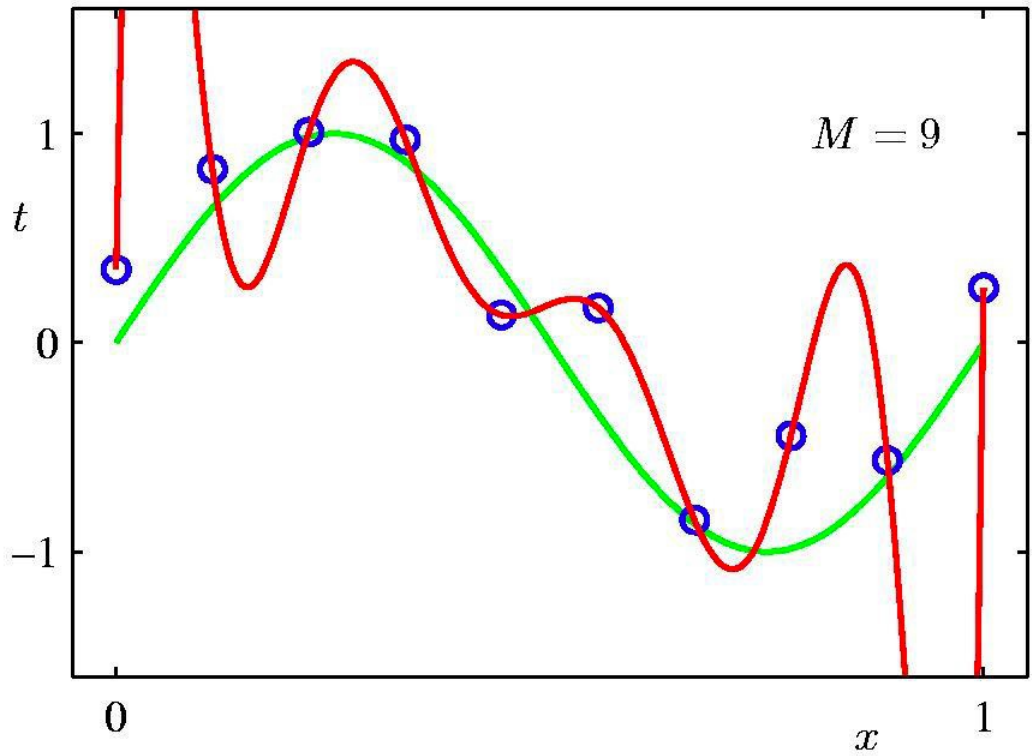
...



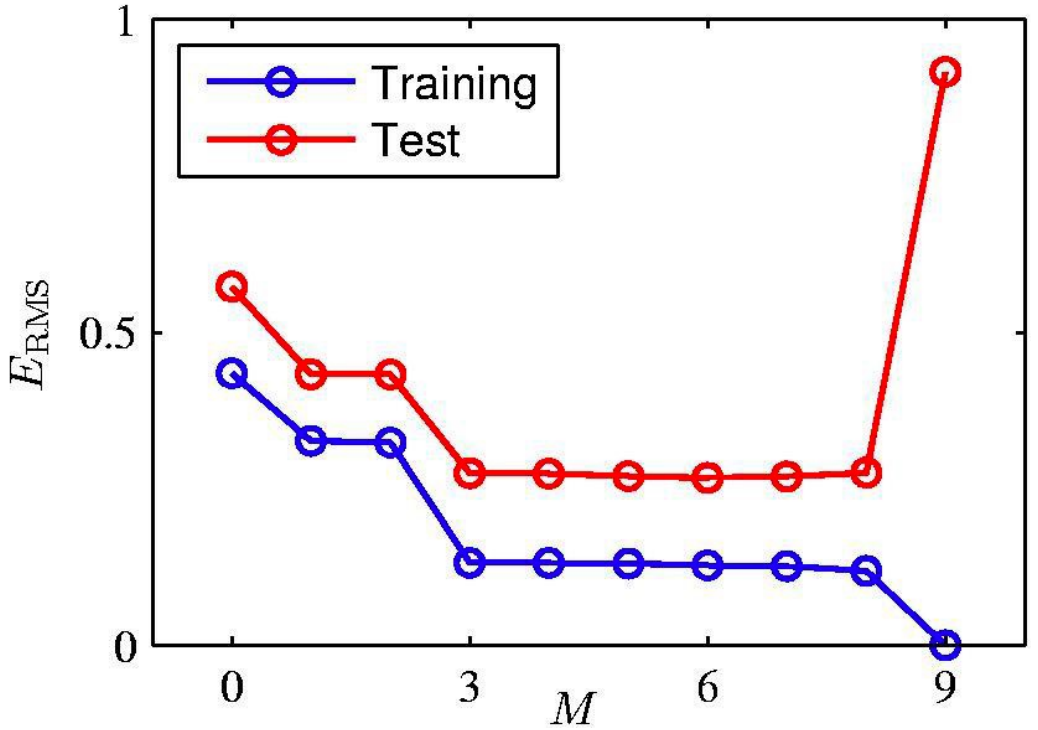
9th order polynomial regression

Overfitting and RMSE

@Ilias Tagkopoulos



- Zero Training Error



Root-Mean-Square (RMS) Error: $E_{\mathrm{RMS}} = \sqrt{2E(\mathbf{w}^\star)/N}$

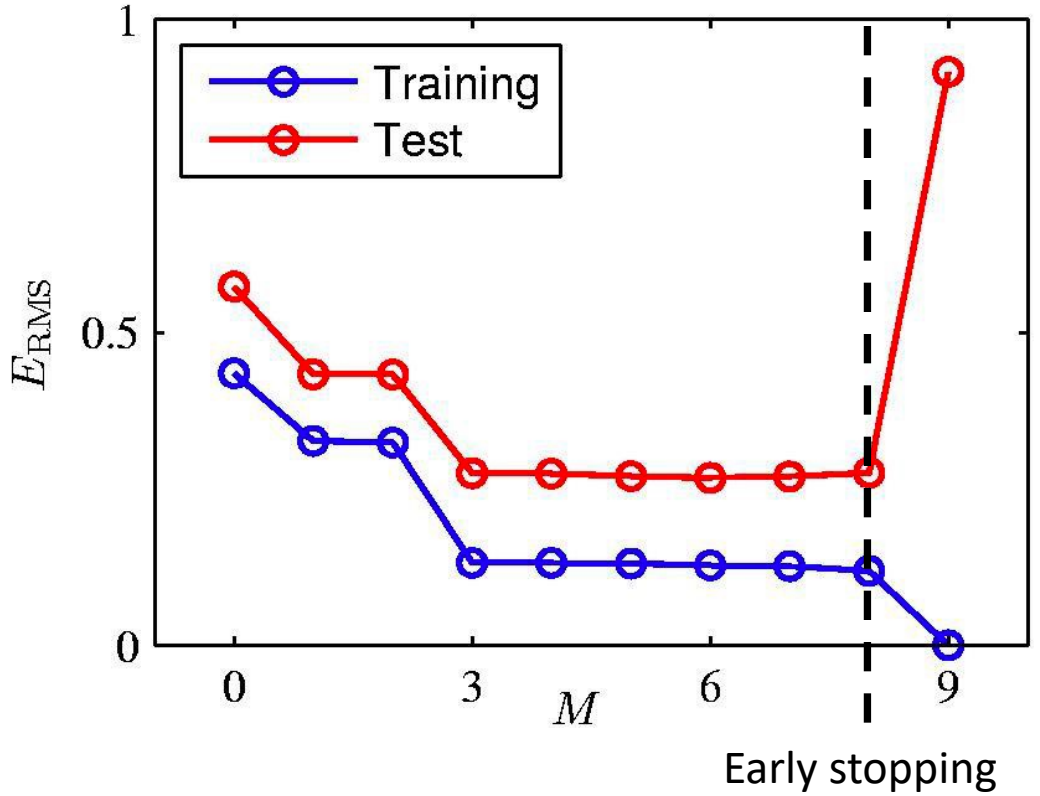
Overfitting(coefficients)

@Ilias Tagkopoulos

	$M=0$	$M=1$	$M=3$	$M=9$
$w_0^\star$	0.19	0.82	0.31	0.35
$w_1^\star$		-1.27	7.99	232.37
$w_2^\star$			-25.43	-5321.83
$w_3^\star$			17.37	48568.31
$w_4^\star$				-231639.30
$w_5^\star$				640042.26
$w_6^\star$				-1061800.52
$w_7^\star$				1042400.18
$w_8^\star$				-557682.99
$w_9^\star$				125201.43

Prevent Overfitting

- Cross validation
- Increase sample size for training data
- Remove features
- Early stopping
- Regularization
- Ensembling



Regularization

- Example: Ridge Regression, Lasso
- M : number of samples in D
- N : number of attributes

$$\sum_{i=1}^{M}(y_i-\hat{y}_i)^2=\sum_{i=1}^{M}\left(y_i-\sum_{j=0}^{N}w_j\times x_{ij}\right)^2$$

Cost function for simple linear model

$$\sum_{i=1}^{M}(y_i-\hat{y}_i)^2=\sum_{i=1}^{M}\left(y_i-\sum_{j=0}^{N}w_j\times x_{ij}\right)^2+\lambda\sum_{j=0}^{N}w_j^2$$

Cost function for ridge regression

Penalty = sum of the square of the magnitude of the coefficients, multiplied by the penalty term (λ).

In ridge regression, the loss function is revised by adding a penalty.

$$\text{For some } c>0,\ \sum_{j=0}^{N}w_j^2<c$$

Constraint on Ridge regression coefficients

Minimizing RSS in LR: Optimization Problem

- In Linear Regression, the basic assumption is that with minimizing the RSS, the relationship between input and output can be outlined in the best possible way.

$$\boldsymbol{w} \triangleq \underset{\boldsymbol{w}}{\operatorname{argmin}}\, RSS = \underset{\boldsymbol{w}}{\operatorname{argmin}} \left(\sum_{i=1}^{m} \left(y^{(i)} - \boldsymbol{w}^T \boldsymbol{x}^{(i)} \right)^2 \right)$$

- Remembering how to minimize the RSS:
  1. **Ordinary Lease Squares (OLS) : Method 1 – Analytical approach**
     1. OLS for minimizing the RSS is equivalent to maximizing the log likelihood of the data given the model➔ Lets talk about Maximum Likelihood Estimation for Linear Regression.
  2. **Gradient Descent (GD) : Method 2 – Numerical approach**

Maximum Likelihood for Linear Regression

- Maximum Likelihood Estimation (MLE) is a probabilistic framework to estimate the parameters of a linear regression model.
- Linear Regression can be formulated as a conditional probability problem. MLE can be used to estimate the supervised learning model parameters.
- Linear regression model parameters can be estimated using negative log likelihood function from MLE.
- The negative log likelihood function is used to derive the least squares solution.

MLE for LR cont.

$Sample: X_1, X_2, \ldots, X_n$

$y: output\ (response\ variable)$

$$y = \alpha + \beta x$$

when residual error $\epsilon = 0$.

$$y = \overbrace{\alpha + \beta x}^{f(x)} + \epsilon$$

Random noise

Model this error following a distribution

$$\epsilon \overset{iid}{\sim} \mathcal{N}(0, \sigma^2)$$

Convert LR to p.d.f

Response variable — Random noise

Given a fixed, non-random sample $X_1, X_2, \dots, X_n$: $\{y = f(x) + \epsilon \longrightarrow \epsilon \overset{iid}{\sim} \mathcal{N}(0, \sigma^2)$

$f(x) = \alpha + \beta x$ — σ^2 = variance

$$For\ a\ fixed\ sample\ X_i, the\ distribution\ of\ Y_i is\ equal\ to: \mathcal{N}(f(X_i), \sigma^2)$$

mean — variance

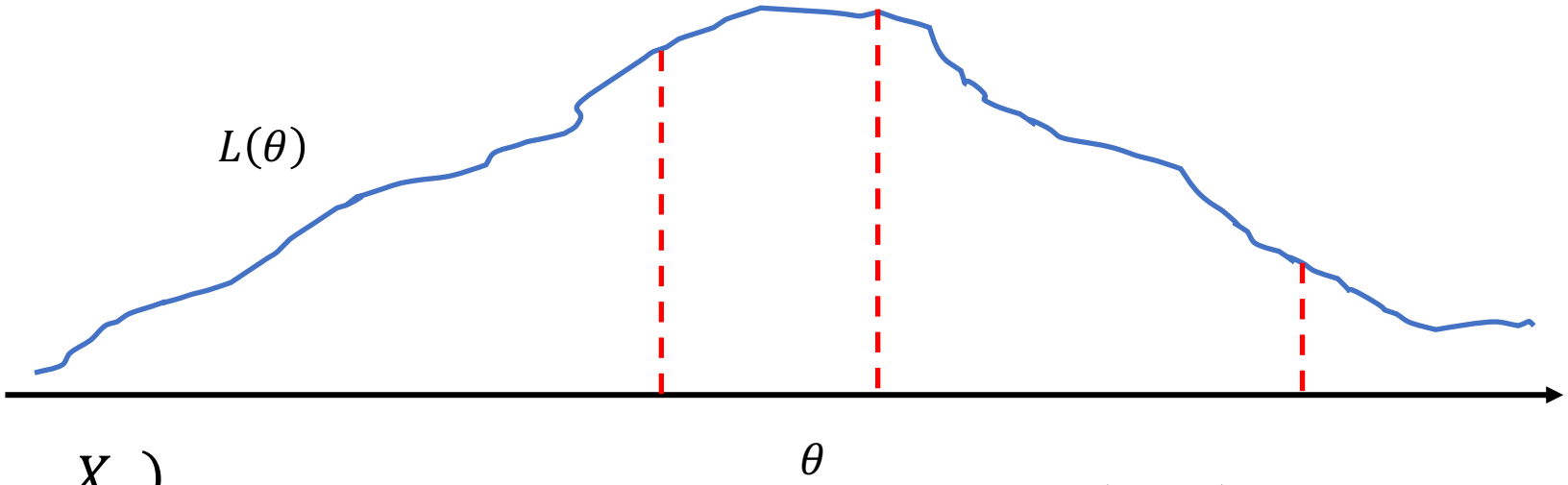
General form of a Probability Density Function (p.d.f):

$$f(y) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y-\mu)^2}{2\sigma^2}} \Rightarrow f(y) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y-f(x_i))^2}{2\sigma^2}}$$

$$f(X) = w^T X \rightarrow \mathcal{N}(w^T X, \sigma^2)$$

Likelihood Function

$L(\theta), where\ \theta\ is\ the\ unknown\ parameter$



$$L(\theta|X_1, \dots, X_n) = f(X_1, \dots, X_n|\ \theta)$$
$$= f(X_i|\ \theta)$$
$$= f(X_1|\ \theta) \dots f(X_n|\ \theta)$$
$$= \prod_{i=1}^{n} f(X_i|\ \theta)$$

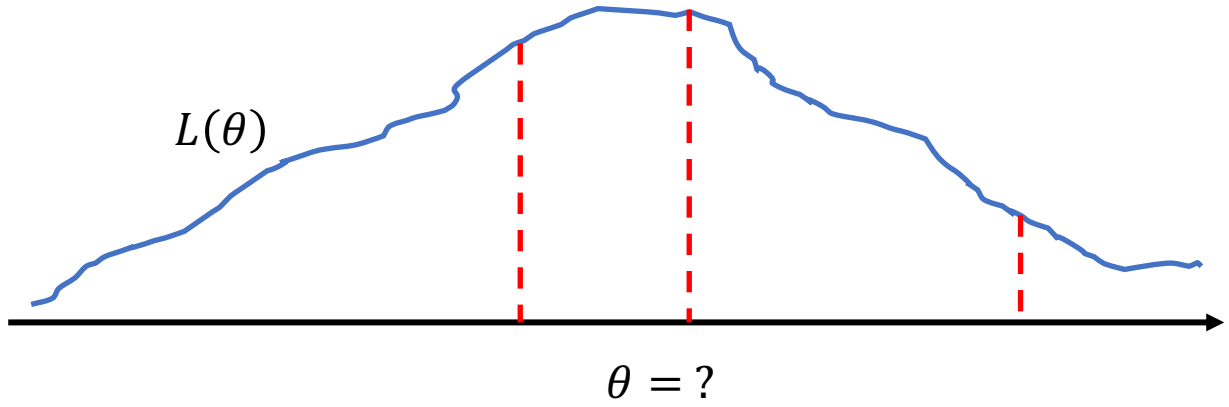
$$\log(a.b) = \log a + \log b$$

$$L(\theta|X_1, \dots, X_n) \cong - log\ L(\theta|X_1, \dots, X_n)$$

Negative Log Likelihood (NLL)

MLE cont.

$$\theta \triangleq argmax_\theta\, p(D|\,\theta)$$
$$\theta \triangleq argmax_\theta \log p(D|\,\theta)$$



N= number of training samples in dataset D

$$L(\theta) \triangleq \log p(D|\,\theta) = \sum_{i=1}^{N} \log p(Y_i|X_i, \theta)$$

$$= \sum_{i=1}^{N} \log\left[\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(Y_i - w^T X_i)^2}{2\sigma^2}}\right]$$

$$= -\frac{N}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{N}(Y_i - w^T X_i)^2$$

Recall p.d.f:

$$f(y) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(y-\mu)^2}{2\sigma^2}}$$

$$\log(a.b) = \log a + \log b$$
$$\log e^x = x$$

MLE cont.

$$L(\theta) \triangleq -\frac{N}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{N}(Y_i - w^T X_i)^2$$

$$RSS(w) \triangleq \sum_{i=1}^{N}(Y_i - w^T X_i)^2 = ||\epsilon||^2$$

sum of squared errors (SSE)

squared norm of residual errors.

$$NLL(\theta) \triangleq -\log f(X_i \mid \theta) = -\sum_{i=1}^{N}\log p(Y_i \mid X_i, \theta)$$

Basis Function Expansion

- Basis expansion $\varphi(x)$ is a class of modelling techniques that can model more complex relationships by augmenting the input features X with some transformations and then use the transformed features in linear models.

$$p(y|x,\theta) = N\left(y|w^T\varphi(x), \sigma^2\right)$$

For example, $\varphi(x)$can be the vector [1,x,…, x^d]

- Some widely used basis functions:
  - $\varphi(x) = X_m$
  - $\varphi(x) = X_j X_k$
  - $\varphi(x) = log(X_j), sqrt(X_j)$