The Machine Learning Problem

Risman Adnan
Telkom University
rismanadnan@telyu

Outline

- The Machine Learning Problem
- Type of Machine Learning
- Simple Polynomial Curve Fitting
- Tools and Frameworks
- Homework #1 :

The Learning Problem

- Automatic discovery of **regularities** in data through computer algorithms and the use of those regularities to take actions such as ***classifying*** the data into categories.

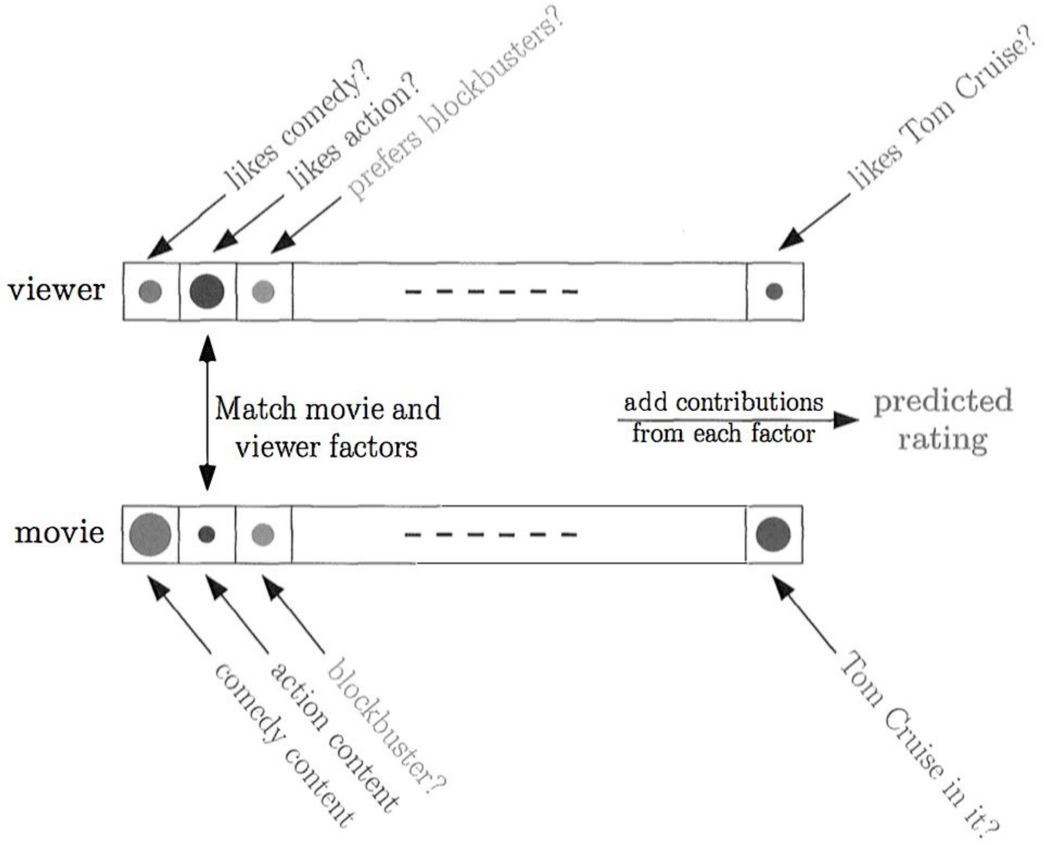
- **The Essence of Machine Learning**:
  - A pattern exists
  - We cannot pin it down mathematically (no analytical solution)
  - We have data on it

- **Initial**: Problems that hard for human – but easy for computer.
- **Current**: Problems that hard for computer– but easy for human.

Example: Netflix

- **Problem**
  - Predict users rating
  - No information about users and movies

- **Dataset**
  - 100,480,507 ratings, 480,189 users, 17,770 movies
  - Training Data: <*user, movie, data of grade, grade*>
  - Test Data 2,817,131: <*user, movie, data of grade, ?*>

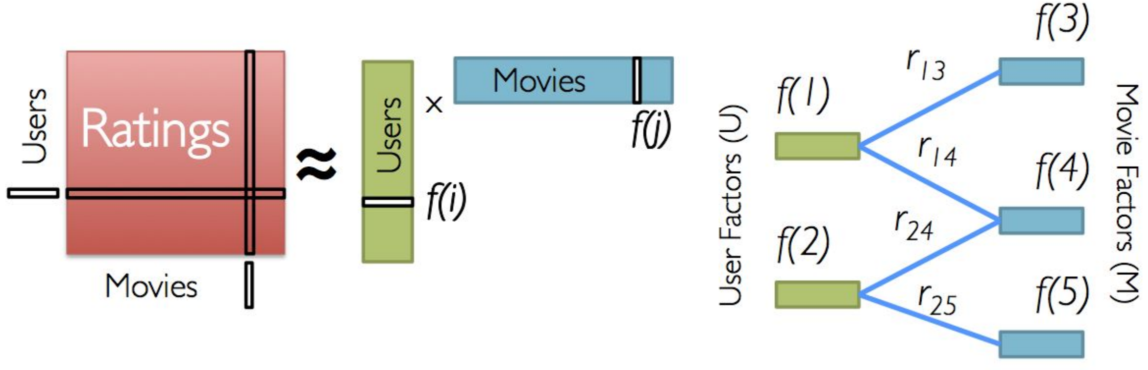


Example: Netflix

Common Techniques:

- Collaborative Filtering
- Matrix Factorization
- Random Initialization
- Iteration to Minimize Error
- Alternating Least Squares

Low-Rank Matrix Factorization:



Iterate:

$$f[i] = \arg\min_{w \in \mathbb{R}^d} \sum_{j \in \mathrm{Nbrs}(i)} \left(r_{ij} - w^T f[j]\right)^2 + \lambda ||w||_2^2$$

Example: MNIST

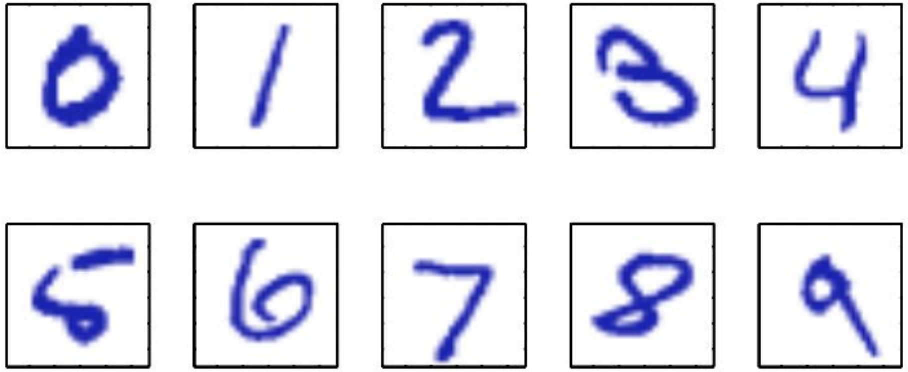
Task

Classify handwritten digits

Build a machine that take x input and produce digit identity 0,..9.

Dataset

MNIST: http://yann.lecun.com/exdb/mnist/

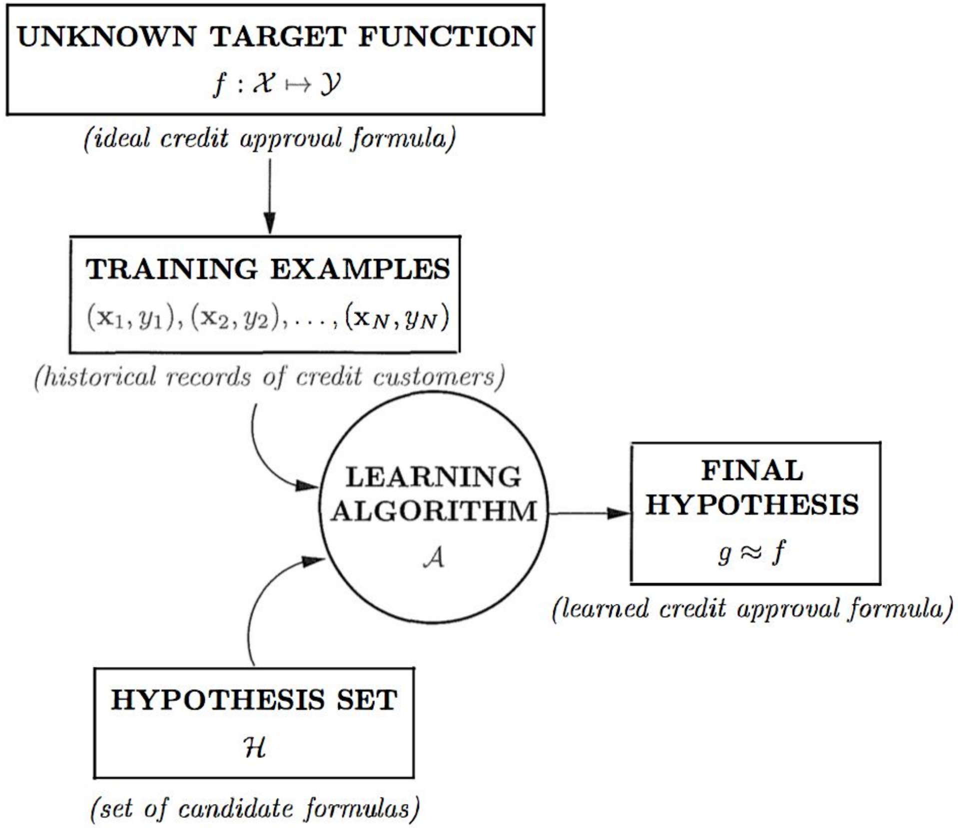


Component of Learning

Metaphor: Credit Card Approval

- Input: **x** (*customer application*)
- Output: *y* (*good/bad customer*)
- Unknown Target Function *f*
- Data (*historical record of credit customers*)
- Hypothesis Set & Final Hypothesis

Learning Model = Hypothesis Set + Learning Algorithm



Machine Learning Paradigm

Supervised Learning: Learning by labeled example

E.g. An email spam detector

We have (**input**, **correct output**), and we can predict (**new input**, **predicted output**)

Amazingly effective if you have lots of data

Unsupervised Learning: Discovering Patterns

E.g. Data clustering

Instead of (**input**, **correct output**), we get (**input, ?**)

Difficult in practices but useful if we lack labeled data

Reinforcement Learning: Feedback & Error

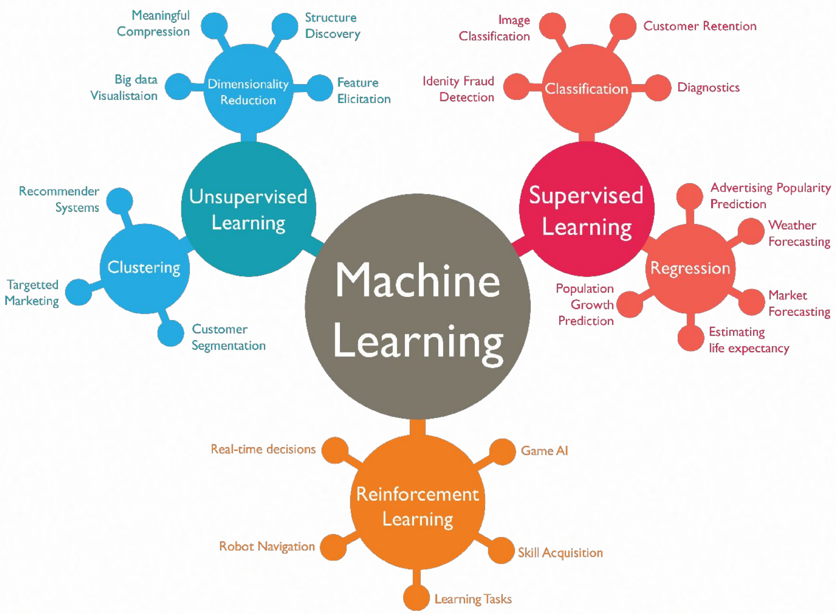
E.g. Learning to play chess

Instead of (**input**, **correct output**), we get (**input**, **only some output**, **grade of this output**)

Works well in some domains, becoming more important

The Landscape

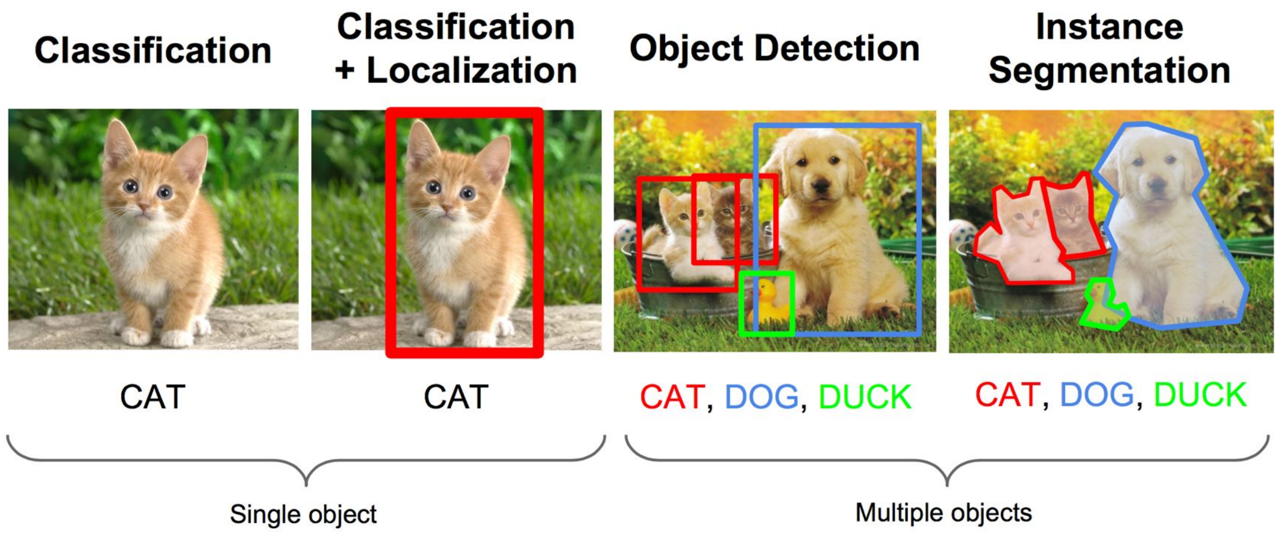
- Theory is Required to Guide Engineering
- Model = Hypothesis Set + Algorithm
- Methods = How to make it works well
- Paradigms = The way we see ML problems



THEORY	TECHNIQUES		PARADIGMS
	MODELS	METHODS	
VC	**Linear**	**Regularization**	**Supervised**
Bias-Variance	**Neural Networks**	**Validation**	**Unsupervised**
Complexity	**SVM**	**Aggregation**	Reinforcement
Bayesian	**Nearest Neighbors**	**Input Processing**	Active
	RBF		Online
	Gaussian Processes		
	SVD		

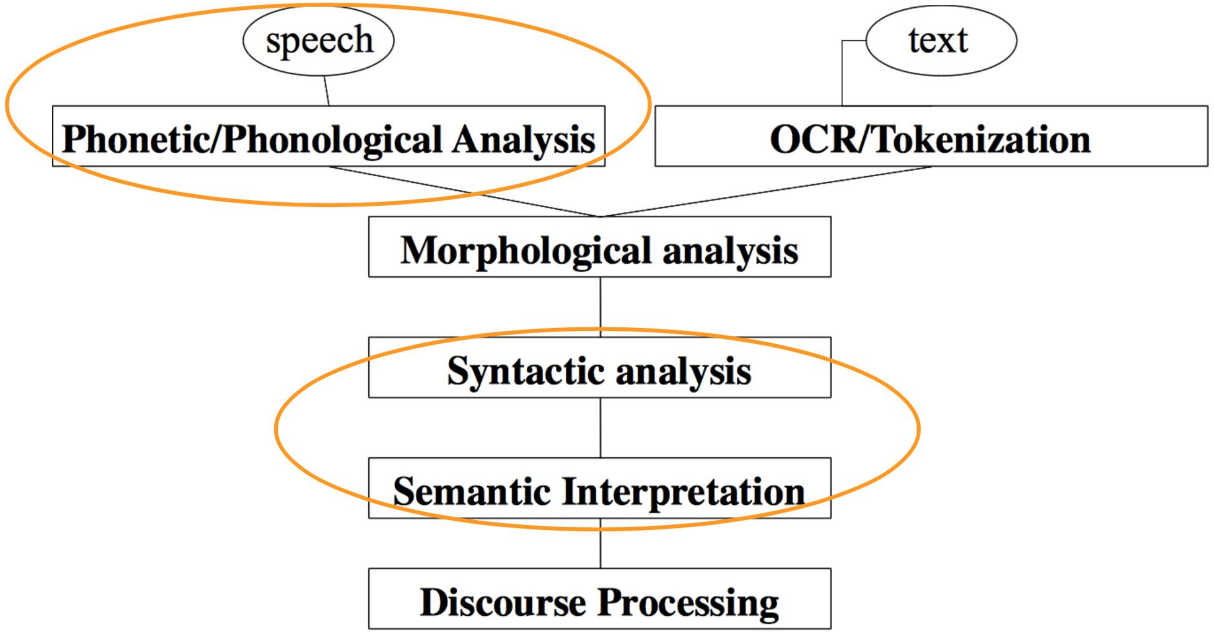
Computer Vision Problem

- Data: Images, Videos
- Pattern: Spatial
- Tasks: Many….



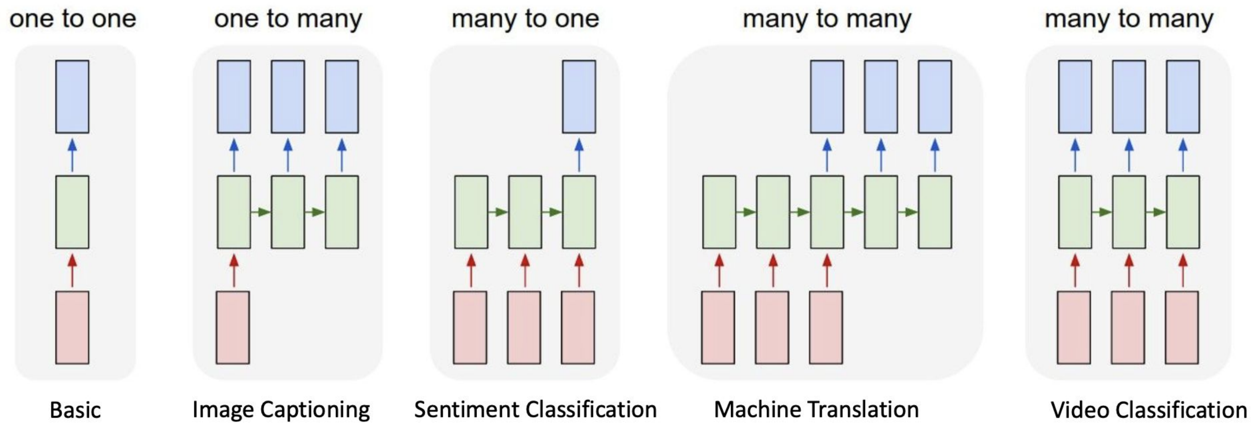
Natural Language Processing Problem

- Data: Speech, Texts
- Pattern: Sequential
- Tasks: Many …

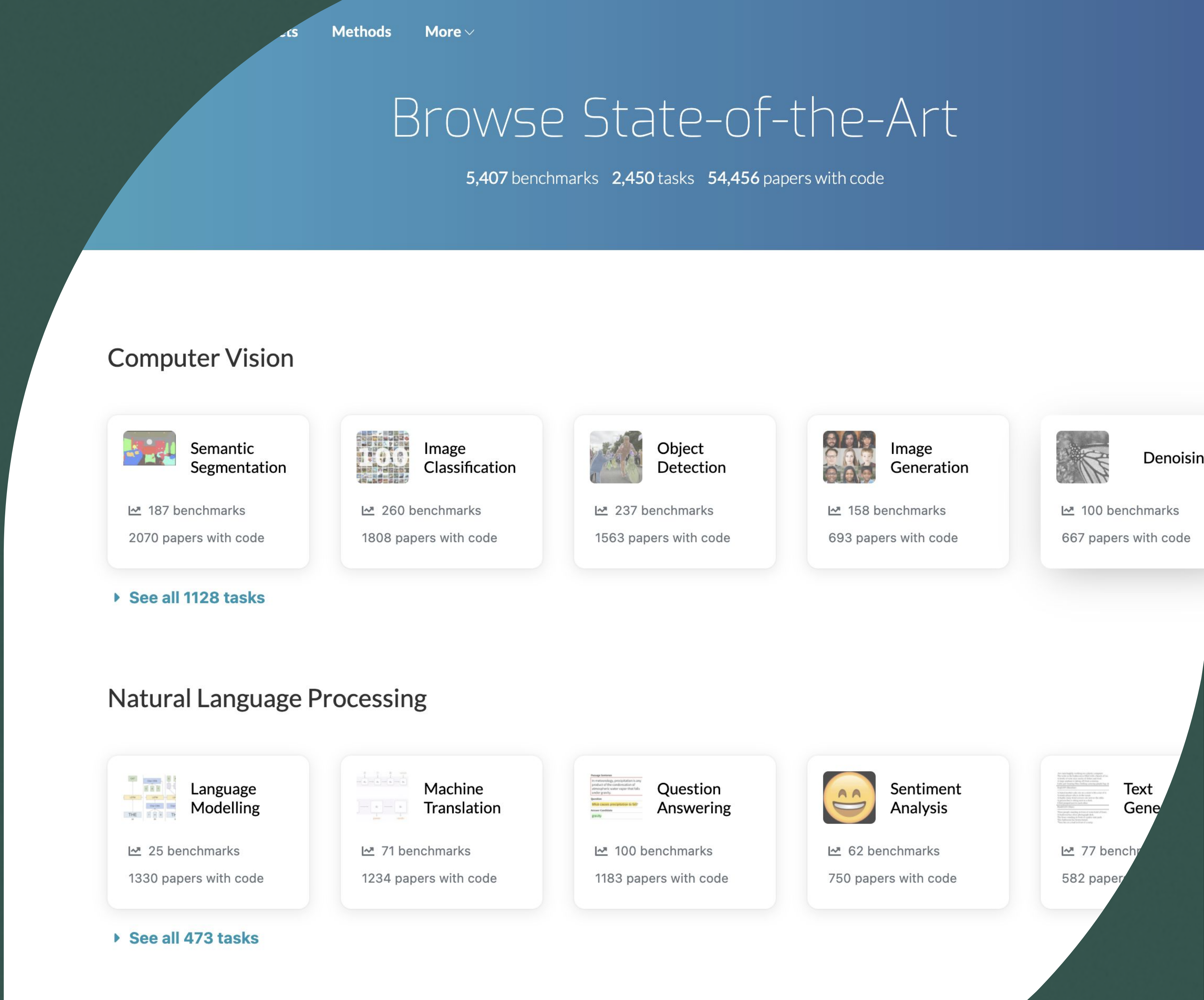


Sequence to Sequence Problem

- Data: Texts, Speech
- Pattern: Sequential
- Tasks: Many …



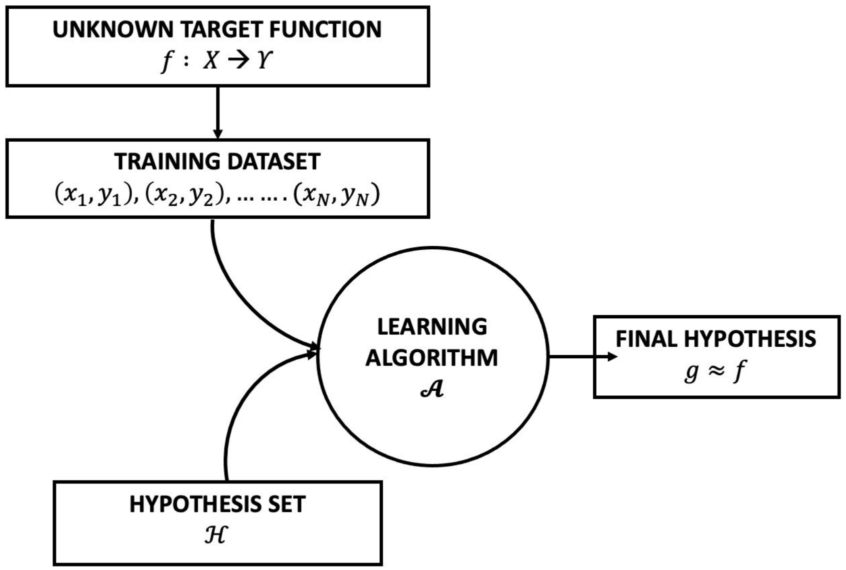
PapersWithCode



Thinking Framework

Discriminative Supervised Learning:

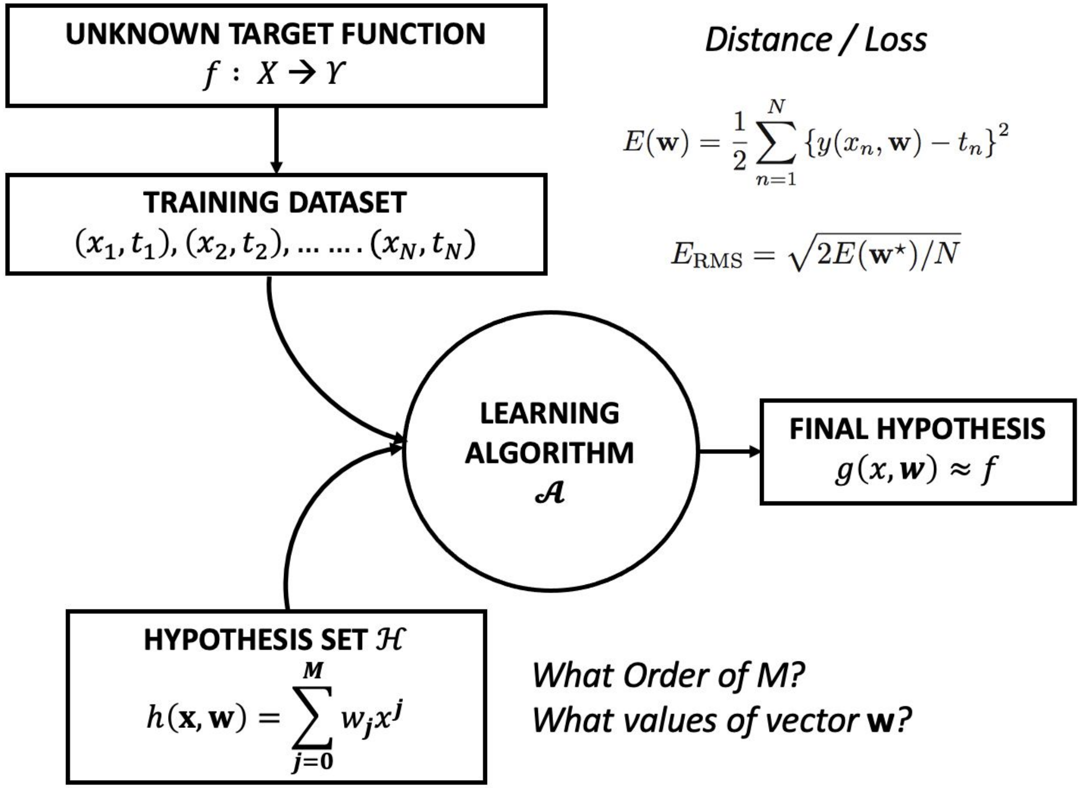
- Problem: Estimate "unknown" function that maps data to labels
- This Problem is Translated to:
  - Classification – If Labels are Categorical
  - Regression – If Labels are Continous
- Approach: Make hypothesis set that potentially has an approximated solution
- Technique: Use algorithm to find an approximated function



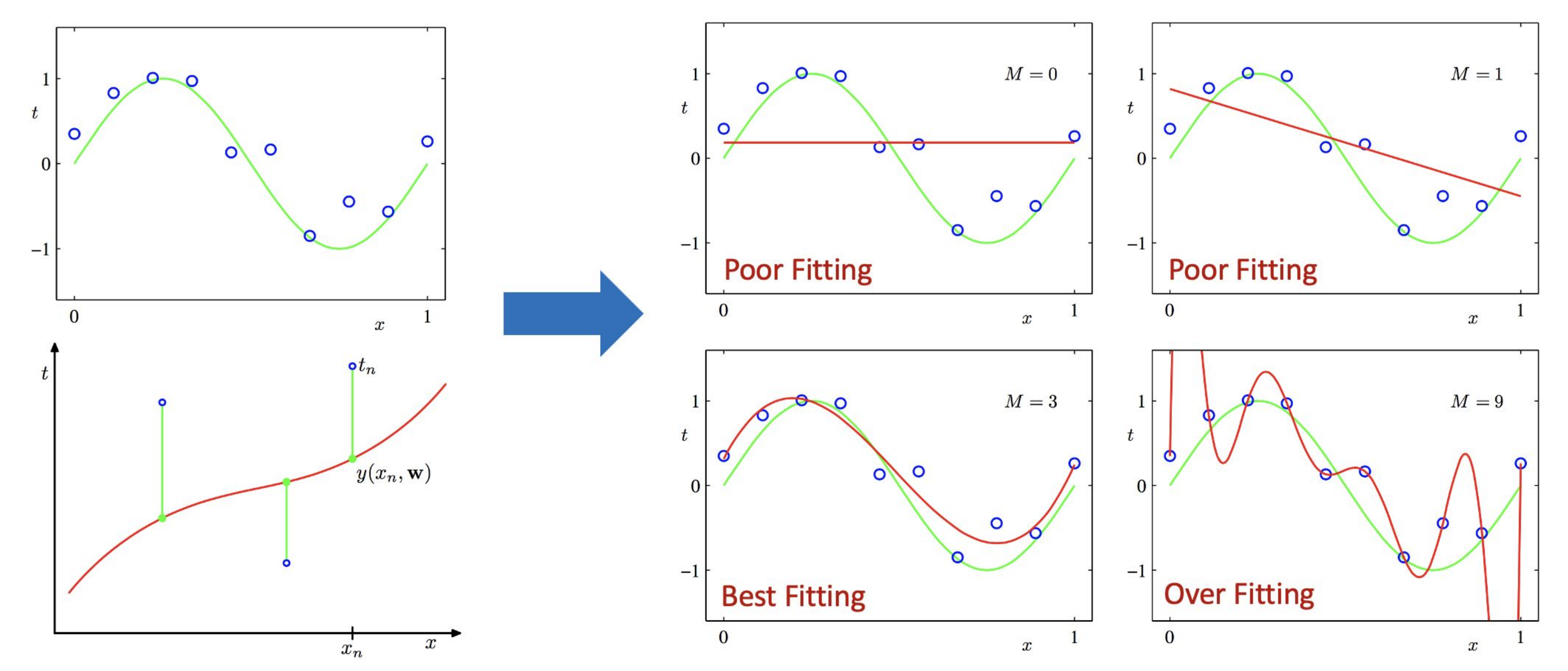
Polynomial Curve Fitting

- Simple Regression Problem
- Linear Model – Polynomial
- Goal: **Good Generalization**
- Learning Algorithms

    - Minimize Square Error Function *E* with GD
    - Or Minimize Root Mean Square (RMS)

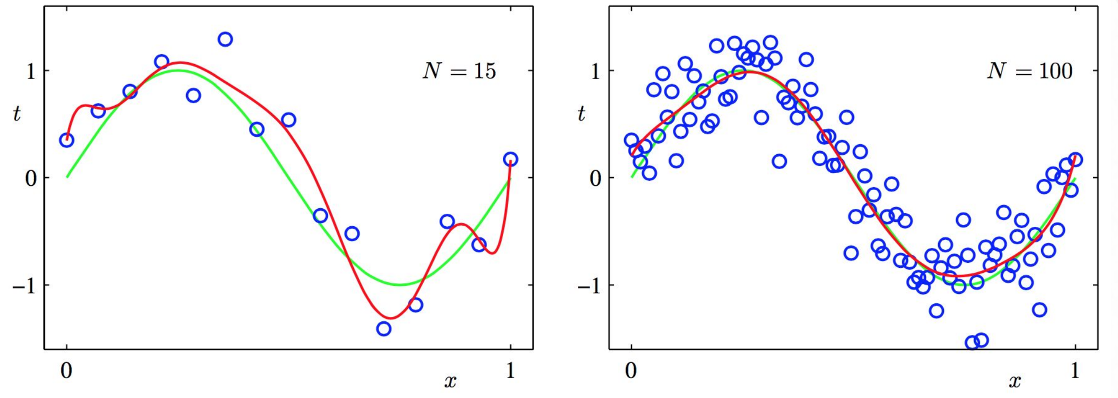


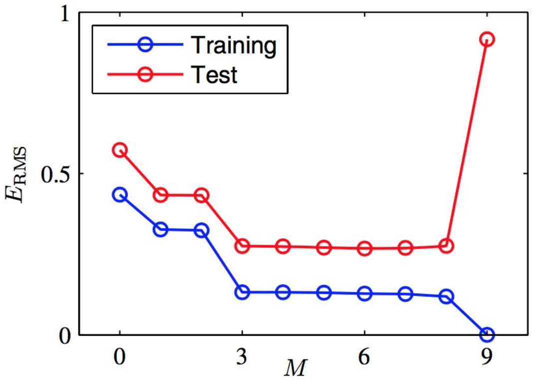
Polynomial Curve Fitting



Polynomial Curve Fitting

- RMS for Equal Footing & Same Scale
- Min Training & Test Error ($3 \le M \le 8$)
- Wild Oscillation for Test Data at $M \geq 9$
- Error = Zero $M \geq 9$ (Over-fitting)
- <u>Add More Dataset Get Less Over-fitting.</u>



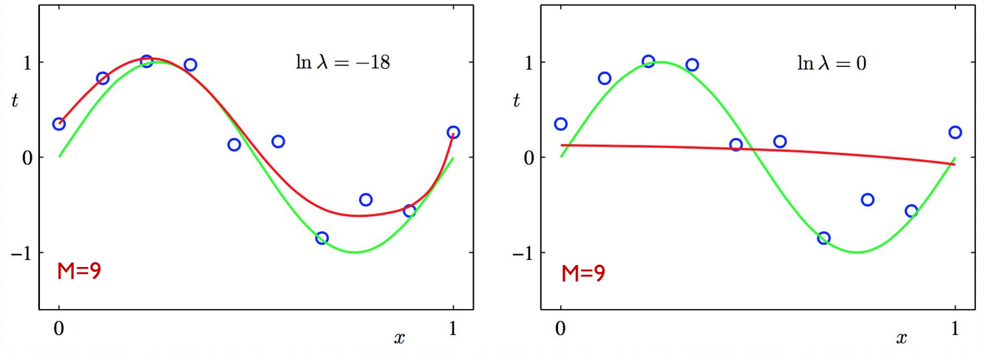


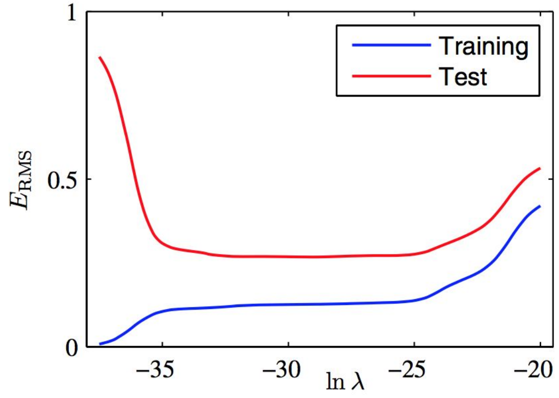
	$M=0$	$M=1$	$M=6$	$M=9$
$w_0^\star$	0.19	0.82	0.31	0.35
$w_1^\star$		-1.27	7.99	232.37
$w_2^\star$			-25.43	-5321.83
$w_3^\star$			17.37	48568.31
$w_4^\star$				-231639.30
$w_5^\star$				640042.26
$w_6^\star$				-1061800.52
$w_7^\star$				1042400.18
$w_8^\star$				-557682.99
$w_9^\star$				125201.43

Polynomial Curve Fitting

- **<u>Regularization</u>** for Overfitting
- Penalty Term to *E* to Regulate Value of Parameters
- Quadratic Regularizer = Ridge Regression

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





	$\ln\lambda = -\infty$	$\ln\lambda = -18$	$\ln\lambda = 0$
$w_0^\star$	0.35	0.35	0.13
$w_1^\star$	232.37	4.74	-0.05
$w_2^\star$	-5321.83	-0.77	-0.06
$w_3^\star$	48568.31	-31.97	-0.05
$w_4^\star$	-231639.30	-3.89	-0.03
$w_5^\star$	640042.26	55.28	-0.02
$w_6^\star$	-1061800.52	41.32	-0.01
$w_7^\star$	1042400.18	-45.95	-0.00
$w_8^\star$	-557682.99	-91.53	0.00
$w_9^\star$	125201.43	72.68	0.01

Tools and Frameworks

- We Will Use Scikit-Learn Framework & Collab
- Simple and Efficient ML Tools (Complete?)
- Important Lib: Pandas, NumPy, and Matplotlib
- **Data Preprocessing with Scikit-Learn**:
  - Standardization, or Mean Removal and Variance Scaling
  - Non-linear Transformation
  - Normalization
  - Binarization
  - Encoding Categorical Features
  - Imputation of Missing Values
  - Generating Polynomial Features
  - Custom Transformers

Homework: House Prediction

Watching Youtube:

- Decision Tree: https://www.youtube.com/watch?v=_L39rN6gz7Y
- Random Forests: https://www.youtube.com/watch?v=J4Wdy0Wc_xQ

Hacking The Codes: https://www.kaggle.com/learn/intro-to-machine-learning

1. How Models Work
2. Basic Data Exploration
3. Decision Tree Model
4. Model Validation
5. Underfitting and Overfitting
6. Random Forest

kaggle

Open Discussion Every Tuesday Nite 19.00PM