UNIVERSITY OF TORONTO | Engineering

Introduction to Machine Learning

1

UNIVERSITY OF TORONTO | Engineering

Machine Learning – Today and the Past

History of AI
Turing Test
Driving Forces

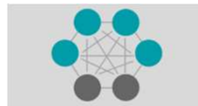
2

History of AI

The Dawn of AI
1950

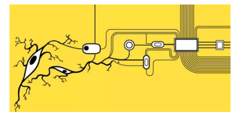
Perceptron
1958

1956
The Dartmouth Conference



The Boltzmann Machine
1983

1997
IBM's Deep Blue



Deep Learning
2006

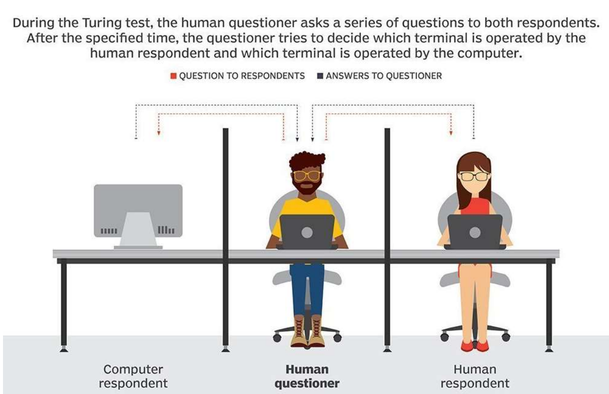
2011
IBM's Watson

AlphaGo
2016

UNIVERSITY OF TORONTO | Engineering

3

Turing Test



Eugene Goostman

- First **pass** on Jun 6, 2014
- 13-year old boy from Odessa, Ukraine

UNIVERSITY OF TORONTO | Engineering

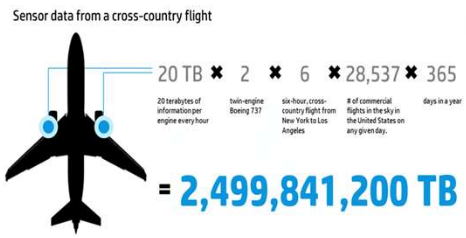
4

Driving Forces

Big Data

More data was created this year than in last **5,000** years
(but only 0.5% was analyzed)

For example:



= **2.5 ZB**

UNIVERSITY OF TORONTO | Engineering

5

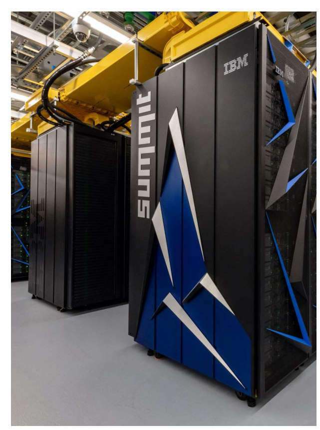
Driving Forces

Big Data

More data was created this year than in last **5,000** years
(but only 0.5% was analyzed)

Computing

If every person on Earth completes one calculation per second, it would take **305 days** to do what Summit can do in **1 second**



UNIVERSITY OF TORONTO | Engineering

6

UNIVERSITY OF TORONTO | Engineering

Basic Concepts of machine learning

ML vs the Traditional Paradigm
Taxonomies of ML
Steps in ML Project

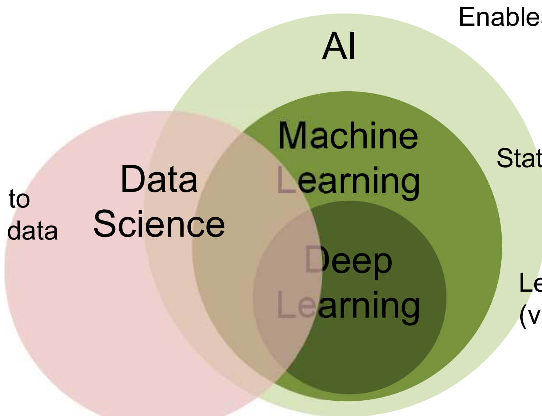
7

AI vs. ML vs…

Enables machine to "think"

Statistical tools to enable AI using data

Mathematical tools to manage & analyse data



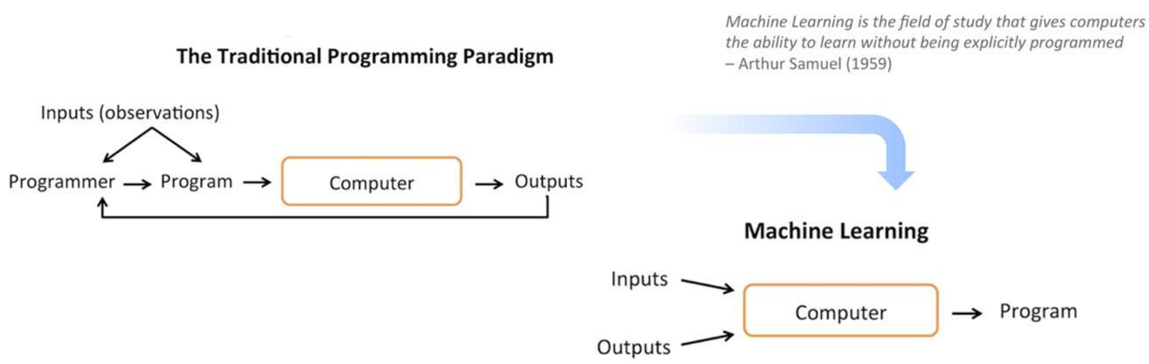
Learning the way humans learn (via multi-neural network architecture)

UNIVERSITY OF TORONTO | Engineering

8

8

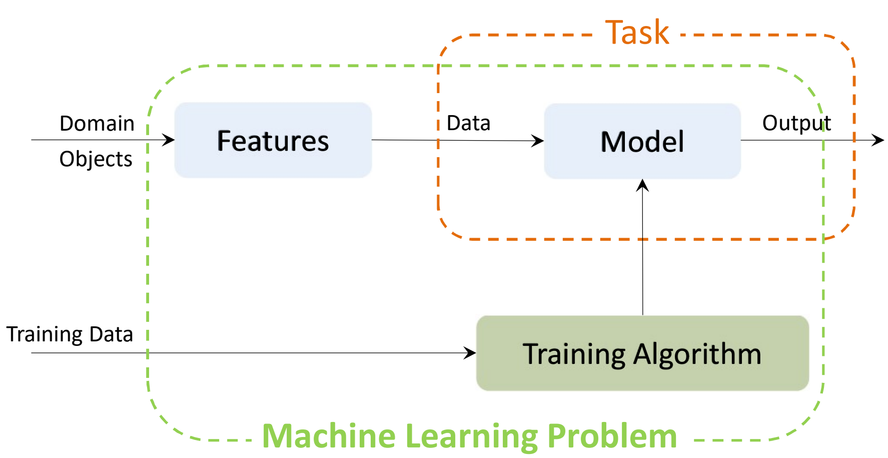
ML vs Traditional Programming



UNIVERSITY OF TORONTO | Engineering

9

Task, Training, Model and Data



UNIVERSITY OF TORONTO | Engineering

10

Taxonomies of ML

Type of Data

- **Supervised ML**
  - Learning by problems and solutions
  - Using labeled examples to predict
- **Unsupervised ML**
  - Learning by problems
  - Explores data to learn hidden patterns
- **Reinforcement learning**
  - Learning by trying
  - Making strategic decisions

Type of Task

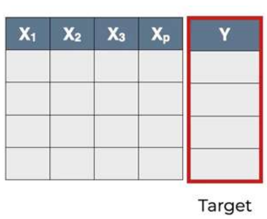
- **Regression**
  - Predict continuous value
- **Classification**
  - Predict discrete value
- **Decision making**
  - Predict the best alternative

UNIVERSITY OF TORONTO | Engineering

11

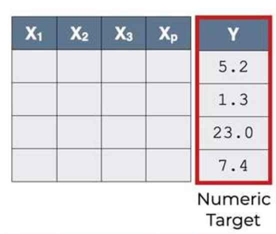
Data and Task

- Data
- Task

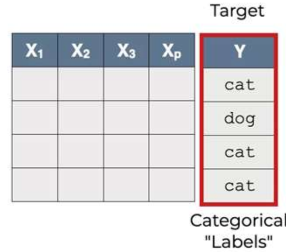


Supervised ML

Unsupervised ML



Regression Task



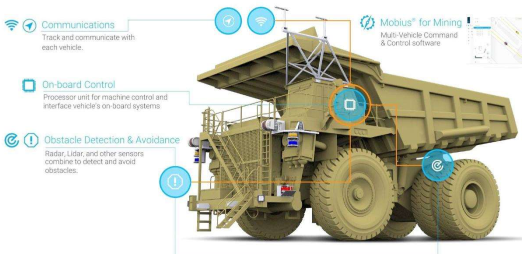
Classification Task

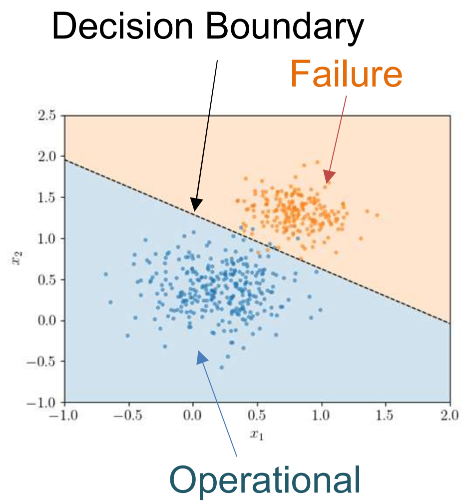
UNIVERSITY OF TORONTO | Engineering

12

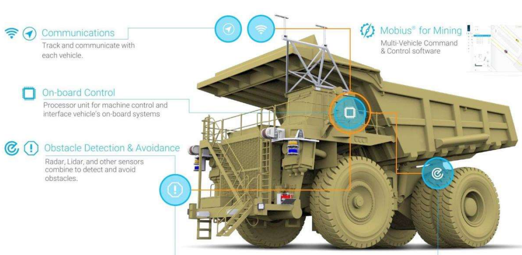
12

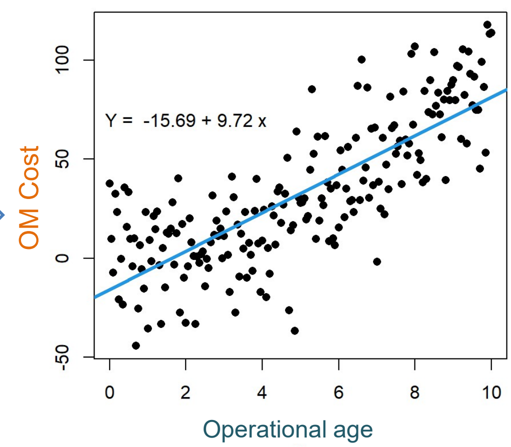
Classification Task





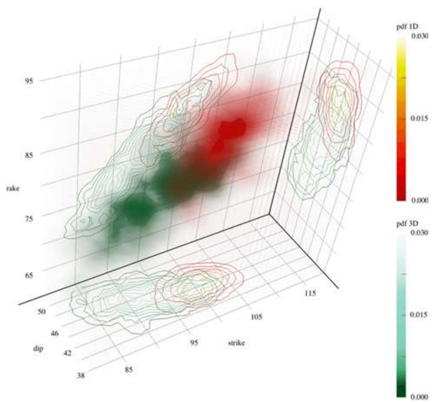
Regression Task





Task types

- Other tasks



dimension reduction

sequential mining

Steps in Machine Learning

1. Define the problem
2. Data Collection — The more, the better
3. Data Preparation — Duplicates, missing data, normalization, split
4. Choose a Model — Nature of tasks
5. Train the Model — Update learnable parameter to reduce loss
6. Evaluate the Model — Loss against unseen data
7. Tune Hyper-parameters of the Model — Learn unlearnables
8. Use the Model to Predict

Machine Learning in Practice

- Preparation of Data
  - One-hot Coding
  - Binning
  - Normalization
  - Standardization
  - Missing data & Imputation
  - Unbalanced data
- Three Sets
  - Training set
  - Validation set
  - Test set
- Selection of Learning Algorithm
  - Explainability
  - # of features and examples
  - Categorical vs. numerical features
  - Nonlinearity of the data
  - Training and Prediction speed
- Performance Evaluation
  - Confusion Matrix
  - Precision, Recall, Accuracy, AUC
- Overfitting and Underfitting

UNIVERSITY OF TORONTO | Engineering

17

UNIVERSITY OF TORONTO | Engineering

Steps in ML Project

Data Preparation
Performance Evaluation

18

Data collection

- Size
  - The more, the better
  - At least 10x more examples than number of trainable parameters
- Quality
  - Noise, even outliers
  - Missing values
  - Features
- Labeling
  - Direct vs. derived

19

Data Preparation

- **1-Hot Coding: Categorical** to **numerical**

$$\text{Red} = [1,0,0] \qquad\qquad \text{Red} = 1$$
$$\text{Yellow} = [0,1,0] \quad \textbf{vs.} \quad \text{Yellow} = 2$$
$$\text{Green} = [0,0,1] \qquad\qquad \text{Green} = 3$$

- **Binning: Numerical** to **categorical**

$$\text{Age} \in [0,5) \quad \rightarrow \quad \text{Bucket} = 1$$
$$\text{Age} \in [6,12) \quad \rightarrow \quad \text{Bucket} = 2$$
$$\text{Age} \in [12,20) \quad \rightarrow \quad \text{Bucket} = 3$$

Data Preparation

- **Normalization**: **Range** to **[0,1]**

$$\bar{x} = \frac{x - \min\{x_1, \dots, x_n\}}{\max\{x_1, \dots, x_n\} - \min\{x_1, \dots, x_n\}}$$

  - Improves learning speed by preventing gradient of large range dominate the gradient descent

- **Standardization**: **Distribution** to **N(0, 1)**

$$\hat{x} = \frac{x - \mu\,\{x_1, \dots, x_n\}}{\sigma\{x_1, \dots, x_n\}}$$

UNIVERSITY OF TORONTO | Engineering

21

Data Preparation

- Normalization vs. Standardization

Normalization	Standardization
	Feature distribution is close to normal
No outliers	Some outliers
Usually recommended over standardization	

UNIVERSITY OF TORONTO | Engineering

22

Data Preparation

- Missing Features and Data Imputation
  - Replacing with an average of the feature

$$\tilde{x} = \frac{1}{M}\sum_{i}^{N} x_i$$

	col1	col2	col3	col4	col5
0	2	5.0	3.0	6	NaN
1	9	NaN	9.0	0	7.0
2	19	17.0	NaN	9	NaN

mean() →

	col1	col2	col3	col4	col5
0	2.0	5.0	3.0	6.0	7.0
1	9.0	11.0	9.0	0.0	7.0
2	19.0	17.0	6.0	9.0	7.0

UNIVERSITY OF TORONTO | Engineering

23

Data Preparation

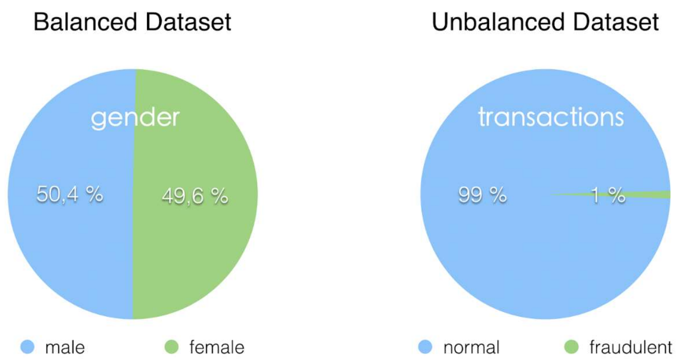
- Other Data Imputation Methods
  - Most frequent value
  - 0 or any other constant
  - kNN
  - Regression
  - Extrapolation and interpolation

UNIVERSITY OF TORONTO | Engineering

24

Data Preparation

- Unbalanced Datasets



UNIVERSITY OF TORONTO | Engineering

25

25

Data Preparation

- Unbalanced Datasets
  1. Carefully select performance metrics

     e.g., Accuracy vs Precision+Recall
  2. Resampling
     1. Under-sampling
     2. Over-sampling
     3. SMOTE (Synthetic Minority Over-sampling Technique)

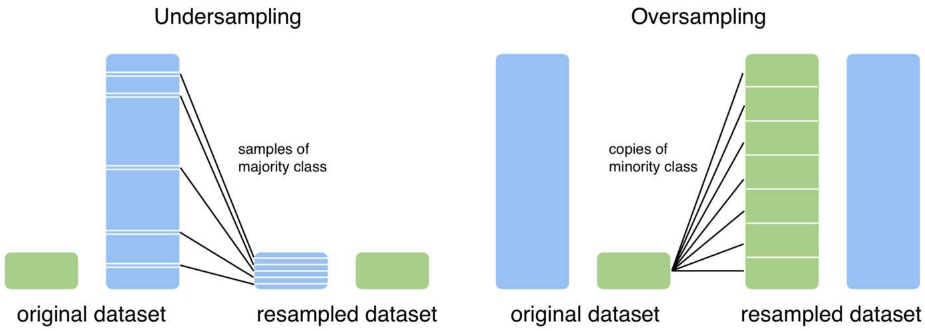
UNIVERSITY OF TORONTO | Engineering

26

26

Data Preparation

- Unbalanced Datasets



Three Sets

- Training Set
  - To build the model
- Validation Set
  - To tune hyper-parameters
- Test Set
- Rule of Thumb
  - **70%:15%:15%** for **training** : **validation** : **test**

Selecting Learning Model

- Explainability
- # of features and examples
- Categorical vs. numerical features
- Nonlinearity
- Training speed
- Prediction speed

UNIVERSITY OF TORONTO | Engineering

29

ML Models

- Supervised ML
  - Linear Models
    - Linear Regression
    - Logistics Regression
    - Support Vector Machine
  - Non-linear Models
    - Naive Bayes Method
    - Decision Trees
    - Neural Networks
- Unsupervised ML
  - Principle Component Analysis
  - Clustering
  - Sequential Patterns

UNIVERSITY OF TORONTO | Engineering

30

Recommendations for Learning Algorithm

	Recommended	Not recommended
Explainability	kNN, **linear regression**, **decision tree**	NN, Kernel-methods
Large data set (feature/sample)	**NN**	SVM
Categorical	**NN**, **Logistic regression**, SVM, **Decision tree**	Linear regression
Non-linearity	**NN**, Kernel-methods	Linear regression, SVM
Training speed	**Regression**, **Decision tree**	NN
Prediction speed	**Regression**, **NN**	kNN, RNN

Back to Steps in Machine Learning

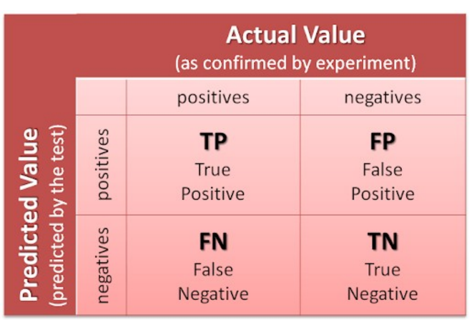
1. Define the problem
2. Data Collection
3. Data Preparation
4. Choose a Model
5. Train the Model
6. Evaluate the Model
7. Tune Hyper-parameters of the Model
8. Use the Model to Predict

Performance Evaluation

- Classification Tasks
  - Confusion matrix
  - Accuracy
  - Precision
  - Recall (sensitivity)
  - F1-score
  - AUC (Area Under ROC Curve)
- Regression Tasks
  - MAE
  - MSE
  - R^2

Classification Tasks

- Confusion Matrix



Predicted	Dove	Woodpecker	Robin	Crow	Starling	Sparrow	Finch	Goldfinch
Dove	11	0	1	0	0	1	0	0
Woodpe	0	9	0	1	1	0	0	0
Robin	0	2	13	0	0	1	0	0
Crow	1	0	0	9	0	0	0	0
Starling	0	0	1	0	12	1	1	0
Sparrow	0	1	2	0	0	13	1	0
Finch	1	0	0	0	0	0	6	1
Goldfinc	0	0	0	0	0	0	2	8

Evaluation of Binary Classification

– Accuracy

$$A = \frac{\#\ of\ correct\ predictions}{\#\ of\ all\ predictions}$$

$$= \frac{TP+TN}{TP+TN+FP+F}$$

– Recall (Sensitivity)

$$R = \frac{\#\ of\ correctly\ predicted\ positives}{\#\ of\ actual\ positives}$$

$$= \frac{TP}{TP+FN}$$

– Precision

$$P = \frac{\#\ of\ correctly\ predicted\ positives}{\#\ of\ all\ predicted\ positives}$$

$$= \frac{TP}{TP+}$$

– F1 score

$$F1 = \frac{2}{\frac{1}{Precision} + \frac{1}{Recall}}$$

$$= 2\frac{Precision \cdot Recall}{Precision + Recal}$$

UNIVERSITY OF TORONTO | Engineering

35

Evaluation of Binary Classification

	Actual Positive	Actual Negative	
Predicted Positive	**TP**=100	**FP**=10	$\hat{\mathbf{P}}$=110
Predicted Negative	**FN**=5	**TN**=50	$\hat{\mathbf{N}}$=55
	P=105	**N**=60	n=165

Recall

Precision

F1 score

Accuracy

UNIVERSITY OF TORONTO | Engineering

36

Evaluation of Multi-Type Classification

Predicted		Dove	Woodpecker	Robin	Crow	Starling	Sparrow	Finch	Goldfinch	TP	FP	FN	Precision	Recall	F1 score
	Dove	11	0	1	0	0	1	0	0						
	Woodpe	0	9	0	1	1	0	0	0						
	Robin	0	2	13	0	0	1	0	0						
	Crow	1	0	0	9	0	0	0	0						
	Starling	0	0	1	0	12	1	1	0						
	Sparrow	0	1	2	0	0	13	1	0						
	Finch	1	0	0	0	0	0	6	1						
	Goldfinc	0	0	0	0	0	0	2	8						

UNIVERSITY OF TORONTO | Engineering

37

Evaluation of Multi-Type Classification

Predicted		Dove	Woodpecker	Robin	Crow	Starling	Sparrow	Finch	Goldfinch	TP	FP	FN	Precision	Recall	F1 score
	Dove	11	0	1	0	0	1	0	0	11	2	2	0.846	0.846	0.846
	Woodpe	0	9	0	1	1	0	0	0	9	2	3	0.818	0.750	0.783
	Robin	0	2	13	0	0	1	0	0	13	3	4	0.813	0.765	0.788
	Crow	1	0	0	9	0	0	0	0	9	1	1	0.900	0.900	0.900
	Starling	0	0	1	0	12	1	1	0	12	3	1	0.800	0.923	0.857
	Sparrow	0	1	2	0	0	13	1	0	13	4	2	0.765	0.867	0.813
	Finch	1	0	0	0	0	0	6	1	6	2	4	0.750	0.600	0.667
	Goldfinc	0	0	0	0	0	0	2	8	8	2	1	0.800	0.889	0.842

UNIVERSITY OF TORONTO | Engineering

38

Regression Task

- MAE (Mean Absolute Error)

$$MAE = \frac{1}{n}\sum |y_i - \hat{y}|$$

- MSE (Mean Squared Error)

$$MSE = \frac{1}{n}\sum (y_i - \hat{y})^2$$

- R^2 (Coefficient of Determination)

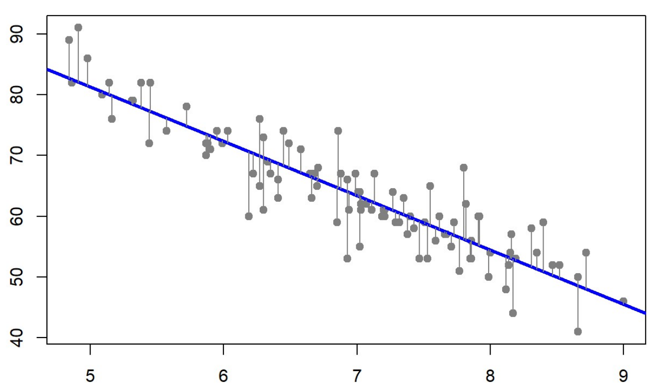
$$R^2 = 1 - \frac{\text{Unexplainable Variation}}{\text{Total Variation}} = 1 - \frac{\sum (y_i - \hat{y})^2}{\sum (y_i - \bar{y})^2}$$

UNIVERSITY OF TORONTO | Engineering

39

Regression Task

- MAE (Mean Absolute Error) and MSE (Mean Squared Error)



$$MAE = \frac{1}{n}\sum |y_i - \hat{y}|$$

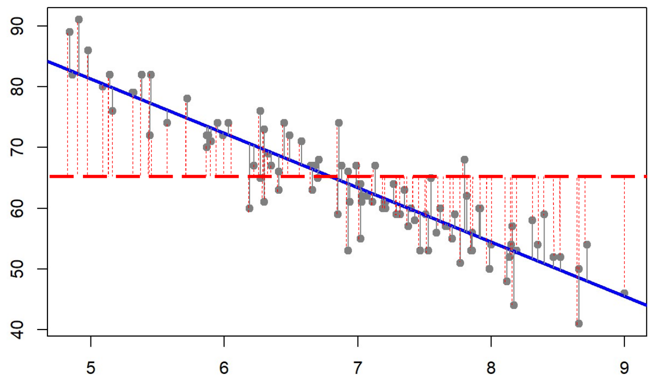
$$MSE = \frac{1}{n}\sum (y_i - \hat{y})^2$$

UNIVERSITY OF TORONTO | Engineering

40

Regression Task

- R^2



Sum of the squared residuals:

$$SS_{res} = \sum (y_i - \hat{y})^2$$

Total variability/variation:

$$SS_{tot} = \sum (y_i - \bar{y})^2$$

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

UNIVERSITY OF TORONTO | Engineering

41

Anomaly Detection on Track

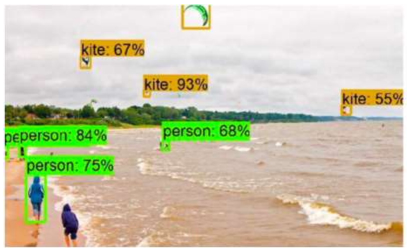
- Subway power-rail may overheat and cause defects
  - 6.5 Km light metro line
  - In May 2017, a train was damaged during operation
  - Power rail anomaly can be caused by high temperature
- IR camera is used to identify anomalies, but the video must be viewed manually to detect the anomalies

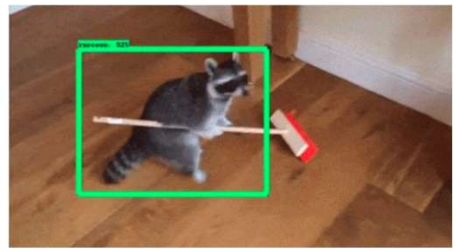
UNIVERSITY OF TORONTO | Engineering

42

Anomaly Detection on Track

- Using TensorFlow Object Detection API & Microsoft Visual Object Tagging Tool to detect anomalies automatically





UNIVERSITY OF TORONTO | Engineering

43

Data and Preparation

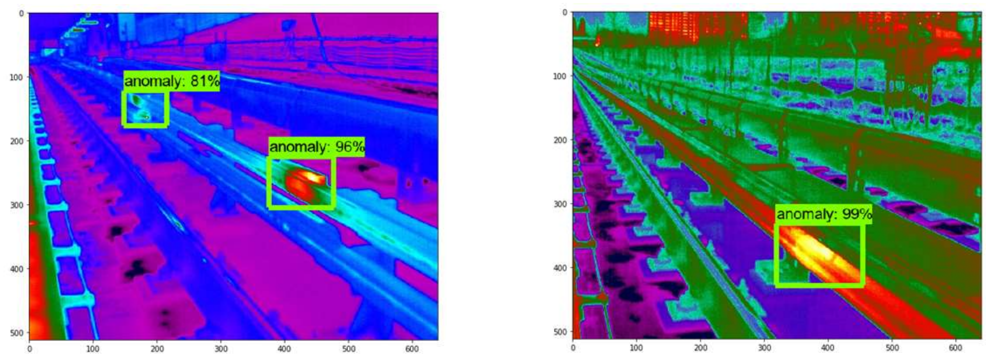
- 2 sets of videos
  - Line 3
  - 1 set has 4 videos with 16,000 frames recorded in December 2017
    - Only 3 videos capture the power rail
    - Partially labelled by TTC with a total 45 anomalies
  - Another set has 8 videos with 10,000 frames recorded in May 2018
    - Only 2 videos capture the power rail
    - Not labelled
  - Winter images were chosen
    - Recorded using a thermal camera in resolution 640X512
    - Further labelling was done by the team to get a total of 668 anomalies

UNIVERSITY OF TORONTO | Engineering

44

Anomaly Detection on Track

- ANN using Tensorflow Object Detection API



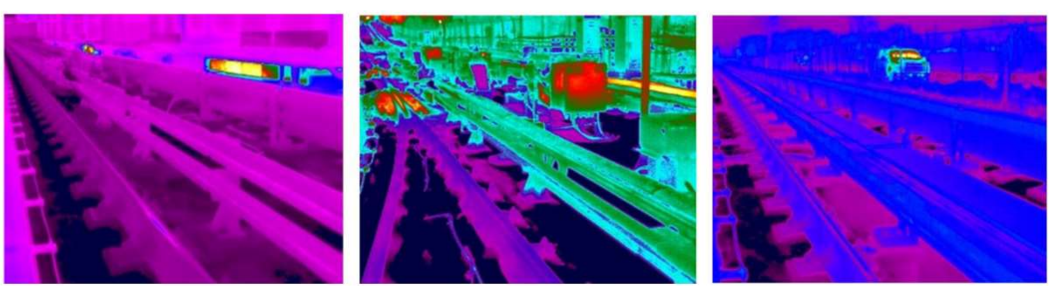
True Positives

UNIVERSITY OF TORONTO | Engineering

45

Anomaly Detection on Track

- Tricky cases



UNIVERSITY OF TORONTO | Engineering

46

Performance

- Balanced Tune-up

Method	Precision	Recall
Shallow AutoEncoder	47%	76%
Robust AutoEncoder	51%	88%
Isolation Forest	64%	94%
SVM	61%	93%

- Recall Maximization

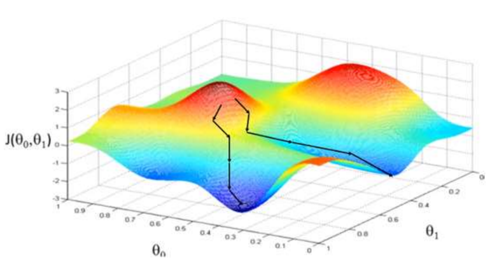
Method	Precision	Recall
Isolation Forest	52%	**100%**
SVM	53%	**100%**

UNIVERSITY OF TORONTO | Engineering

47

Gradient Descent for ML

- The most used learning algorithm




```
Repeat until convergence {
```

$$\theta_{j+1} \leftarrow \theta_j - \alpha \frac{\partial}{\partial \theta_j} \mathrm{MSE}(\theta)$$

```
}
```

$$MSE = \frac{2}{n} \sum (y_i - \hat{y})^2$$

$$\frac{\partial}{\partial \theta_j} \mathrm{MSE}(\theta) = \frac{2}{n} \sum_{i=1}^{n} (\theta^T \cdot x_i - y_i)[x_i]_j$$

UNIVERSITY OF TORONTO | Engineering

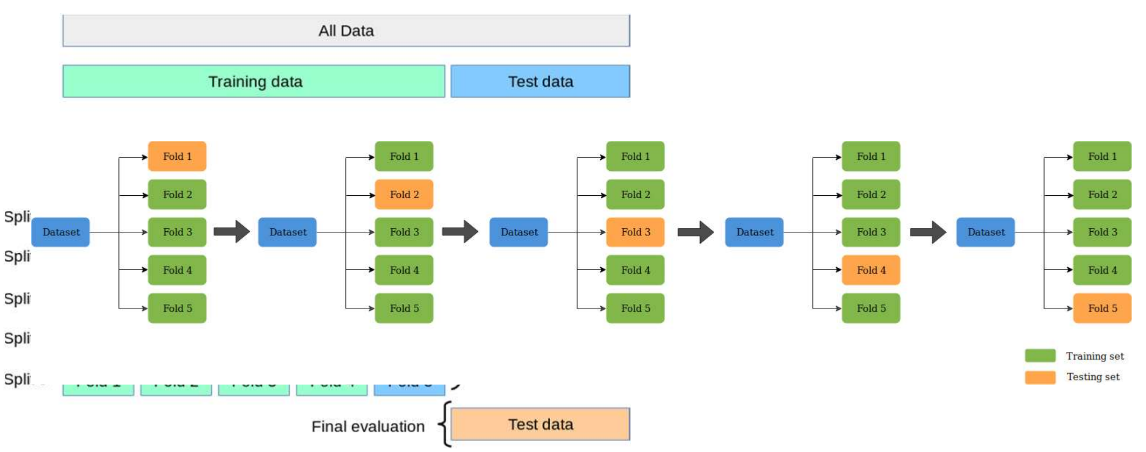
48

Cross Validation

- Cross validation (CV) usually means k-fold cross validation
  - Useful when data set is too small
  - In k-fold, every sample is used for training as well as for testing
  - Recommended in hyper-parameter tune up

- In k-fold CV
  - Data set is split into k subsets of equal size
  - The holdout method is repeated k times
  - Each time, (k-1) sets are used for training and 1 set for testing

k-Fold Cross Validation



UNIVERSITY OF TORONTO | Engineering

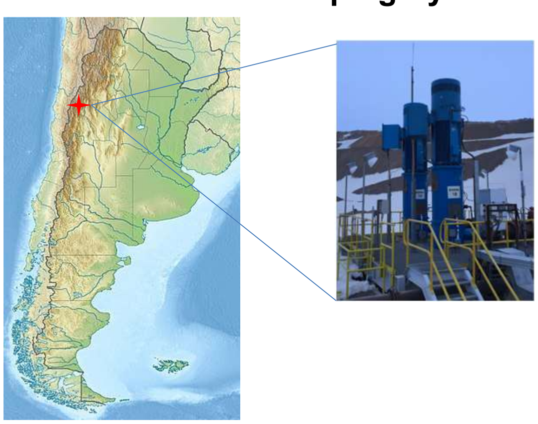
Model Training and Performance Improvement

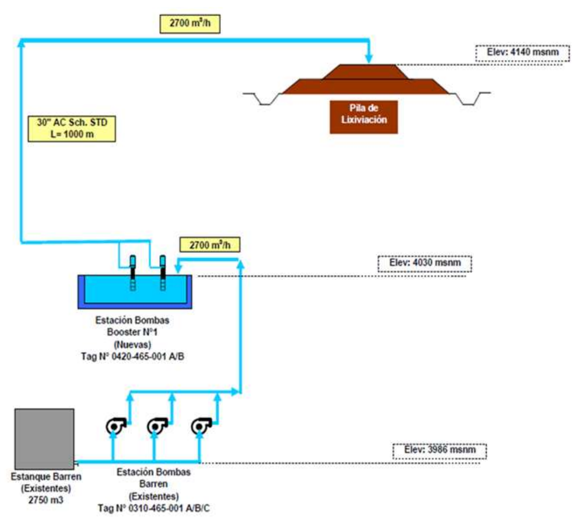
A Case Study
Over-fitting and Under-fitting
Hyper-parameter tune up
Combining learners

51

Case 1: Linear regression

Veladero Barren Pumping System





UNIVERSITY OF TORONTO | Engineering

52

Case 1: Linear regression

Situation

- Two pumps running full time with sensor data available
- Over the course of project, additional data including pressure, and any significant down time were added.
- For better understanding of the data, we sought to determine the relationships between different variables.

Objective

- Anomaly detection before the accident happens

UNIVERSITY OF TORONTO | Engineering

53

Case 1: Linear regression

Steps

Preprocessing → Data Transforming → Exploratory Data Analysis → Model Construction

Preprocessing:
- Missing values
- Outliers

Data Transforming:
- Format transforming
- Data splitting
- Seasonality

Exploratory Data Analysis:
- Correlation matrix
- Bivariate analysis
- Multivariate analysis

Model Construction:
- Regression models

The variables can be divided into three categories
- Performance variables: Discharge pressure and flow
- Control variables: Speed
- Monitoring variables: Vibration and temperature

We seek to predict the **performance variables** based on the control and monitoring variables.

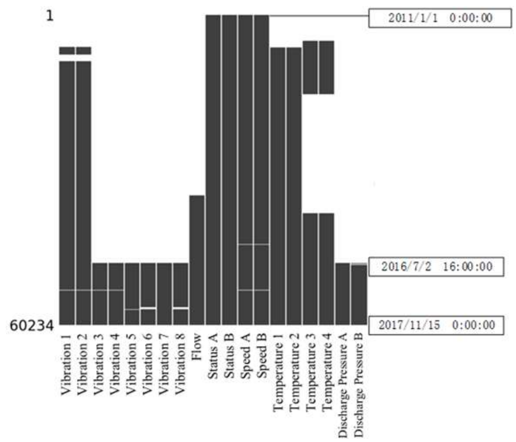
UNIVERSITY OF TORONTO | Engineering

54

Case 1: Linear regression

Missing data

- The records available from 2011/1/1 to 2017/11/15
- Measurements were gradually introduced
- Records only from 2016/07/02 to 2017/11/15 have all the variables



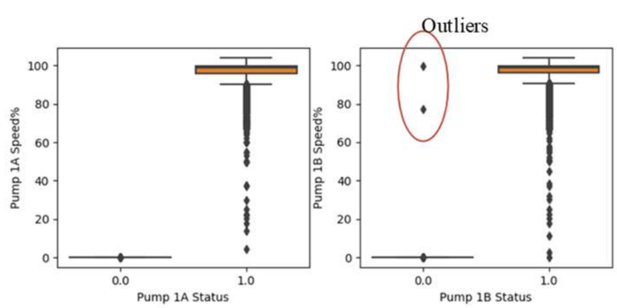
UNIVERSITY OF TORONTO | Engineering

55

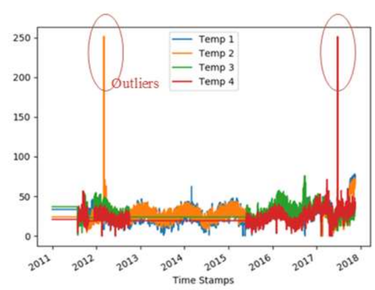
Case 1: Linear regression

Outliers

- Some obvious outliers can be excluded by univariate visualization.
- Also, some extremely high temperatures are clearly unreasonable.



The speeds of different status.



The temperatures

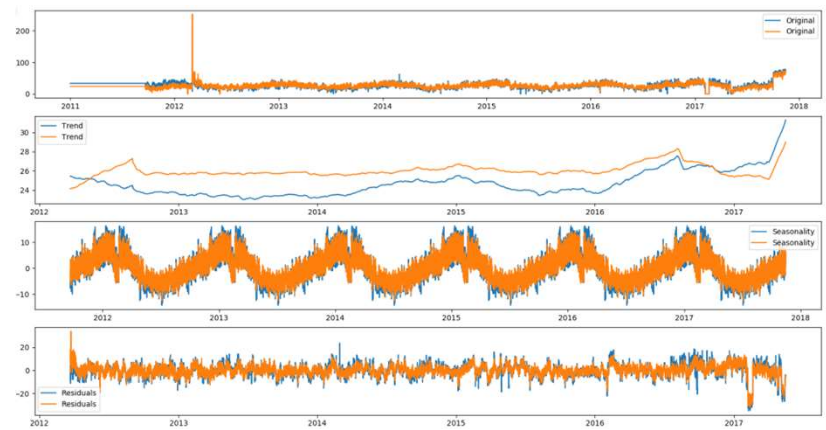
UNIVERSITY OF TORONTO | Engineering

56

Case 1: Linear regression

Seasonality

- The residuals of the temperature are better options in the analysis.

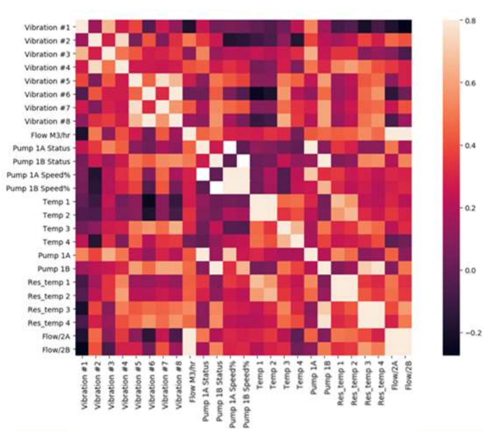


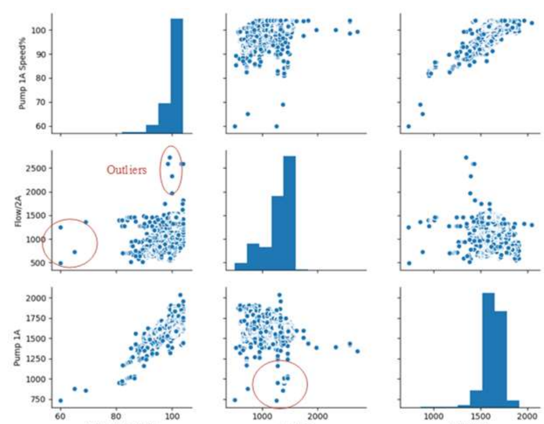
UNIVERSITY OF TORONTO | Engineering

57

Case 1: Linear regression

Correlation & Scatter Plot





- Strong connection between **speed** & **pressure**

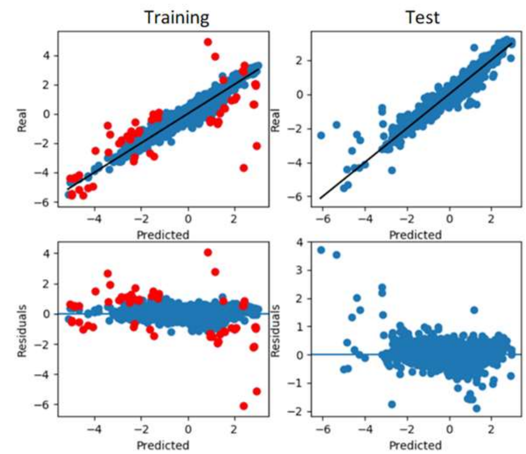
UNIVERSITY OF TORONTO | Engineering

58

Case 1: Linear regression

Linear Regression

- **Speed ➔ Pressure**
- The R^2 values
  - 0.957 for Pump A
  - 0.952 for Pump B.



Early warning symptom

Symptoms = max(Threshold — Residuals, 0)

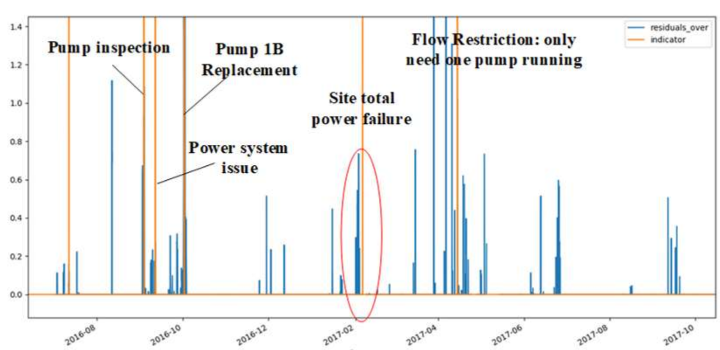
where Residuals =| Pressure(Real) – Pressure(Predicted) |

59

Case 1: Linear regression

Anomaly detection

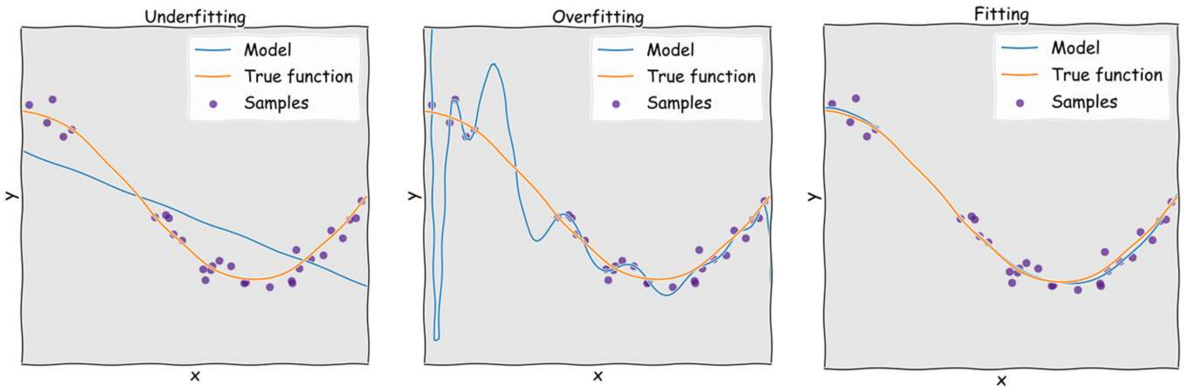
- Threshold is set as 0.45.
- Continuous deviation of the residuals may be treated as an early warning signal.



60

Model Complexity

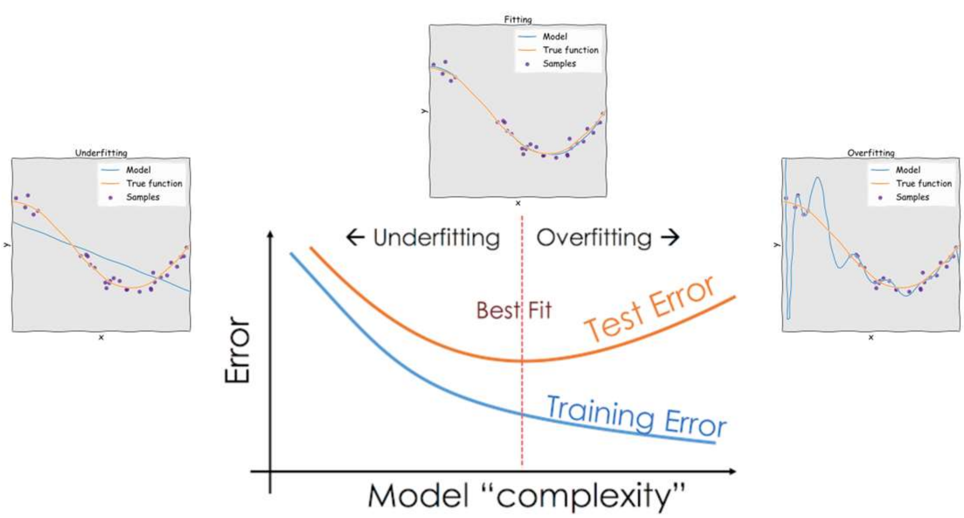
- Under-fitting and Over-fitting



UNIVERSITY OF TORONTO | Engineering

61

Overfitting and Underfitting



UNIVERSITY OF TORONTO | Engineering

62

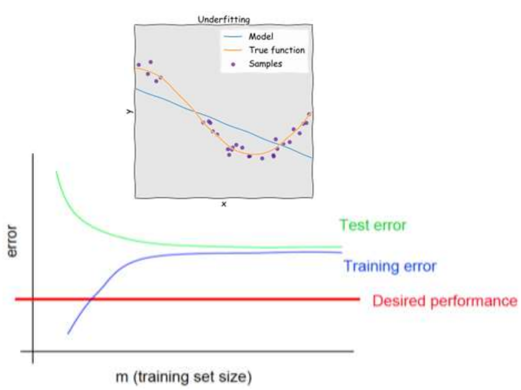
62

Overfitting and Underfitting

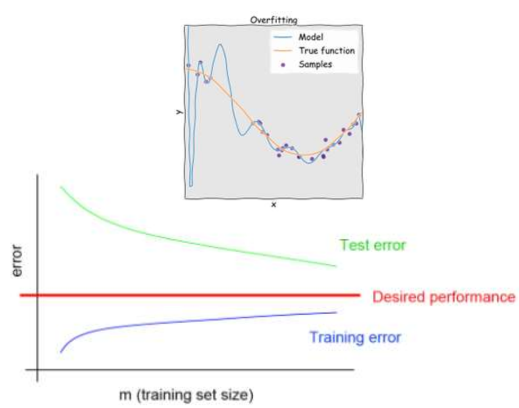
	Overfitting	Underfitting
Model Complexity	Higher	Lower
Degree of Training	More	Less
Size of Data	Smaller	Larger

Overfitting and Underfitting

- How to detect?



Underfitting



Overfitting

Hyper-Parameter Tuning

- Hyper-parameter
  - Not trainable unlike model parameters
  - Quite impactful on the model performance
  - Usually done with cross validation for more reliable performance evaluation
- Manual tuning using experience
- Grid search
  - Commonly used but time consuming
- Random search
- Bayesian optimization
  - Assuming Gaussian process as the objective to be optimized
  - Posterior of GP given samples
  - Relatively easy-to-compute acquisition function

65

Combining Learners

- No single learner performs consistently better than others
- Performance can be improved by combining base learners
- Diversity vs accuracy of base learners
- Two questions:
  - How to generate diverse learners?
  - How to combine the predictions for best results?

66

Generating Diverse Learners

- Different models
- Different hyperprameters
  - # of neurons in DL
  - k in k-nearest neighbor
  - error threshold in decision trees
  - kernel function in support vector machine
- Different input representation
  - Multiple sensors
  - Random subspace (of features)
- Different training sets
  - subsets of training set (bagging or boosting)
  - Partitioning of the training set

Combining Predictions

- Two major approaches
  - Parallel
    - Global: all base learners predict (e.g., voting)
    - Local: some learners are selected, requiring a gating model
  - Serial
    - Subsequent learners are trained where previous ones did not perform

Combining Prediction

- Voting
  - Simplest method to combine classifiers
  - Linear combination of predictions

$$y_i = \sum_j w_j d_{ji} \text{ where } w_j \geq 0, \sum_j w_j = 1$$

  - Classifier combination rules

	C_1	C_2	C_3
d_1	0.2	0.5	0.3
d_2	0.0	0.6	0.4
d_3	0.4	0.4	0.2
Sum	0.2	**0.5**	0.3
Median	0.2	**0.5**	0.4
Minimum	0.0	**0.4**	0.2
Maximum	0.4	**0.6**	0.4
Product	0.0	**0.12**	0.032

Combining Prediction

- Bagging
  - Short for "boostrap aggregating"
  - Base learners are trained with different subsets of training data
  - Base learners should be unstable so that their predictions on test sets are diverse

Steps:

- Given training set X of size N, n instances randomly drawn from X with replacement to make a subset $X_j, j = 1, \dots, L$.
- Base learners d_j are trained with X_j

Combining Prediction

- Boosting
  - In bagging, generating complementary base learners is by chance
  - In boosting, complementary base learners are actively generated by training the next learner on samples previous ones struggled.
  - In the original boosting algorithm (Schapire 1990)
    - Consists of 3 base learners and 3 subsets of samples
    - Train the 3 learner sequentially
    - Not expandible

The original boosting algorithm (Schapire 1990)

- During **training**
  1. Training set is divided into three: X_1, X_2, X_3
  2. Learner d_1 is trained on X_1 and tested on X_2
  3. Make a sample set X_2' with instances in X_2 misclassified by d_1 and the same number of instances in X_2 correctly classified by d_1
  4. Train d_2 on X_2'
  5. Test d_1 and d_2 on X_3 and make a sample set X_3' with instances in X_3 on which d_1 and d_2 disagree
  6. Train d_3 on X_3'
- During **testing**
  1. Given an instance, let d_1 and d_2 predict
  2. If they agree, the prediction is final; if they disagree, the prediction by d_3 is final.

AdaBoost Algorithm (Freund and Schapire 1996)

- Short for "adaptive boosting"
- Use the same training set over and over
- Arbitrary number of base learners
- Basic idea

  Sample instances that were misclassified with a higher probability

Training:
  For all $\{x^t, r^t\}_{t=1}^N \in \mathcal{X}$, initialize $p_1^t = 1/N$
  For all base-learners $j = 1, \ldots, L$
    Randomly draw $\mathcal{X}_j$ from $\mathcal{X}$ with probabilities p_j^t
    Train d_j using $\mathcal{X}_j$
    For each (x^t, r^t), calculate $y_j^t \leftarrow d_j(x^t)$
    Calculate error rate: $\epsilon_j \leftarrow \sum_t p_j^t \cdot 1(y_j^t \neq r^t)$
    If $\epsilon_j > 1/2$, then $L \leftarrow j - 1$; stop
    $\beta_j \leftarrow \epsilon_j / (1 - \epsilon_j)$
    For each (x^t, r^t), decrease probabilities if correct:
      If $y_j^t = r^t$, then $p_{j+1}^t \leftarrow \beta_j p_j^t$ Else $p_{j+1}^t \leftarrow p_j^t$
    Normalize probabilities:
      $Z_j \leftarrow \sum_t p_{j+1}^t; \quad p_{j+1}^t \leftarrow p_{j+1}^t / Z_j$
Testing:
  Given x, calculate $d_j(x), j = 1, \ldots, L$
  Calculate class outputs, $i = 1, \ldots, K$:
    $y_i = \sum_{j=1}^L \left(\log \frac{1}{\beta_j}\right) d_{ji}(x)$

UNIVERSITY OF TORONTO | Engineering

73

73

UNIVERSITY OF TORONTO | Engineering

Thank you!

74