Introduction to Machine Learning

Charu Sharma

Assistant Professor

Machine Learning Lab

IIIT Hyderabad

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

03rd October, 2023

Overview

What is Machine Learning?

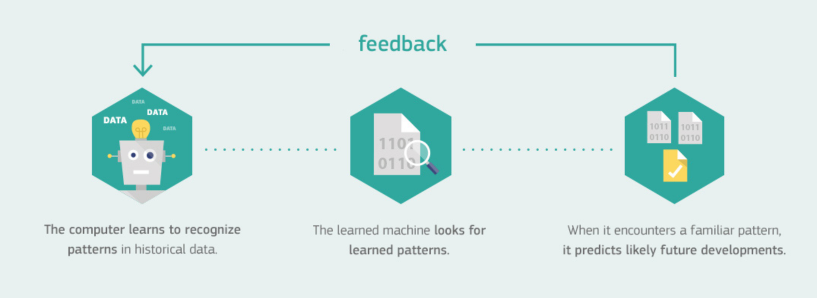
- A field of study that gives computers the ability to learn without being explicitly programmed.
- Well-posed learning problem: A computer program is said to learn from **experience E** with respect to some **task T** and some **performance measure P**, if its performance on T, as measured by P, improves with experience E.
- Inferring knowledge from data. Given a data, it predicts future tasks.

What is Machine Learning?

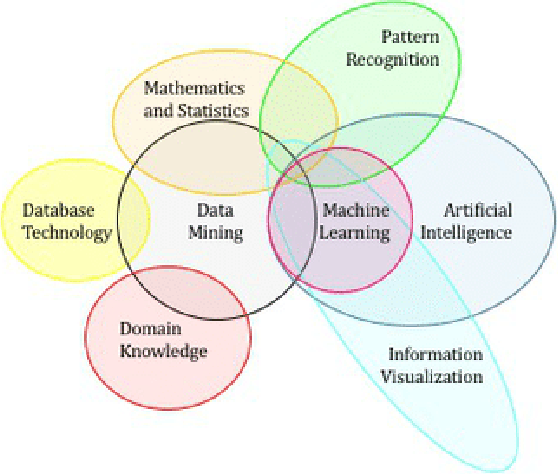
- A field of study that gives computers the ability to learn without being explicitly programmed.
- Well-posed learning problem: A computer program is said to learn from **experience E** with respect to some **task T** and some **performance measure P**, if its performance on T, as measured by P, improves with experience E.
- Inferring knowledge from data. Given a data, it predicts future tasks.

Example: Suppose your email program watches which emails you do or do not mark as spam, and based on that learns how to better filter spam. What is the task T in the setting?
T : to predict an email as spam or not.
E : emails you mark as spam.
P : predicting an email as spam is correct or not.

What is Machine Learning?



Interdisciplinary field



Overview

Why is it important?

- Develop systems that can automatically adapt and customize themselves to individual users. Eg.: spam filtering, personalized news.
- Discover new knowledge from large databases (data mining). Eg.: market basket analysis.
- Ability to mimic human and replace certain monotonous tasks. Eg.: handwritten character recognition.
- Data in many domains is huge. It is impossible for humans to see patterns across so much data. Eg.: processing of images and videos.

Overview

What is Machine Learning?

Why is it important?

Applications

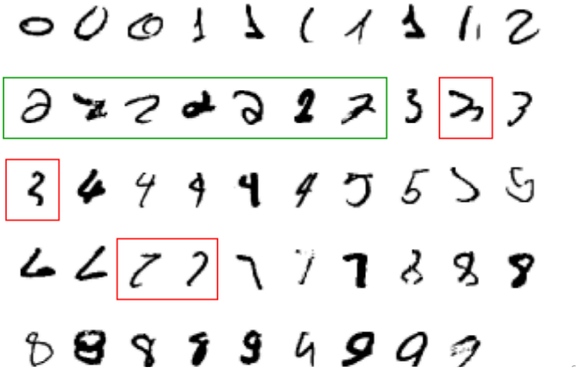
ML Process Flow

ML Applications

Handwritten Digits[1]

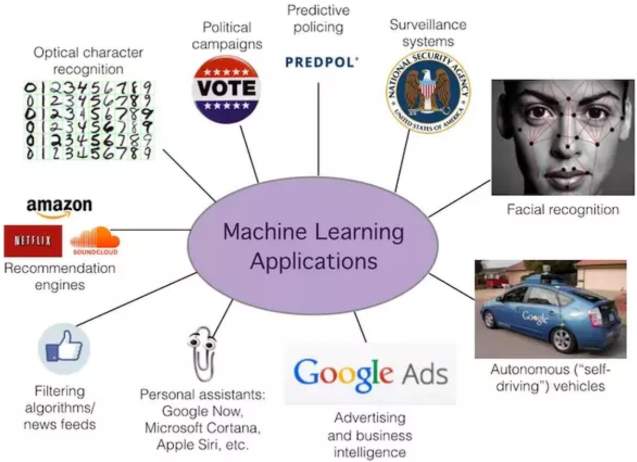
A classic example of a task that requires machine learning:

It is very hard to say what makes a 2



[1]Slide credit: Geoffrey Hinton

Applications[2]



[2] Image Source: Google

Overview

Outline

What is Machine Learning?

Why is it important?

Applications

ML Process Flow
- Gather Data
- Data Preparation
- Machine Learning Algorithms
- Generalization?
- Evaluation Metrics

ML Process Flow

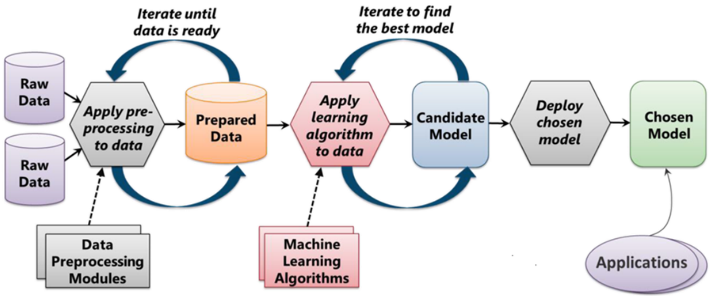


Figure: The learning process[3].

[3]Image Source: Google

Outline

Collect Data

- Collection of data objects/ samples and their features.
- A feature is a property or characteristic of an object.
- A collection of features describe an object.

Columns

Rows

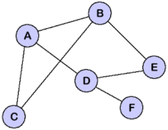
Values

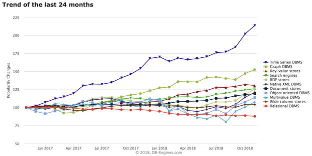
ID	Outlook	Temp	Humidity	Windy	Play Golf
1	Rainy	85	92	False	No
2	Rainy	80	88	True	No
3	Overcast	83	86	False	Yes
4	Sunny	70	80	False	Yes
5	Sunny	68	?	False	Yes
6	Sunny	65	58	True	No
7	Overcast	64	62	True	Yes
8	Rainy	72	95	?	No
9	Rainy	?	70	False	Yes
10	Sunny	75	72	False	Yes
11	Rainy	75	74	True	Yes
12	?	72	78	True	Yes
13	Overcast	81	66	False	Yes
14	Sunny	71	79	True	No

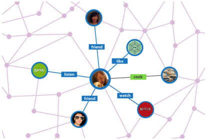
Data Types

Data comes in different sizes and also flavors (types):

- Texts
- Numbers
- Images
- Videos
- Graphs
- Time series
- Social network
- Geographic data







Outline

Data Preparation

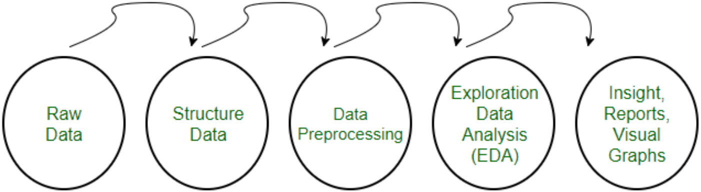


Figure: Data preparation phase[4].

[4]Image Source: Google

Challenges during Data Preparation

Figure: Challenges in handling data[5].

[5] Image Source: Google

Outline

Types: Supervised vs. Unsupervised

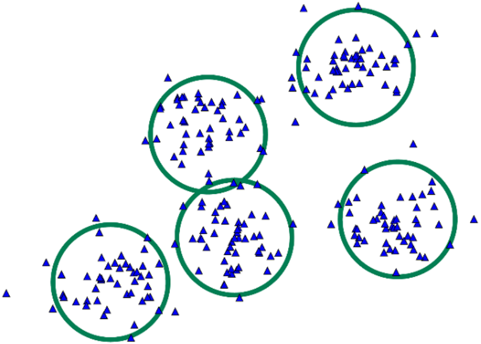
Supervised

- Training data includes both the input and the desired results.
- For some examples the correct results are known and are given in input to the model during the learning process.
- The construction of a proper training, validation and test set is crucial.
- These methods are usually fast and accurate.
- Have to be able to generalize: give the correct results when new data are given in input without knowing a priori the target.

TYPES: SUPERVISED VS. UNSUPERVISED

Unsupervised

- The model is not provided with the correct results during the training.
- Can be used to cluster the input data in classes on the basis of their statistical properties only.
- The labeling can be carried out even if the labels are only available for a small number of objects representative of the desired classes.



Types: Supervised vs. Unsupervised

Given: Training data: (x_1,y_1),..., (x_n,y_n) / $x_i \in \mathbb{R}^d$ and y_i is the label.

sample $x_1 \rightarrow$	x_{11} x_{12} ... x_{1d}	$y_1 \leftarrow$ label
...	... ...	...
sample $x_i \rightarrow$	x_{i1} x_{i2} ... x_{id}	$y_i \leftarrow$ label
...	... ...	...
sample $x_n \rightarrow$	x_{n1} x_{n2} ... x_{nd}	$y_n \leftarrow$ label

Types: Supervised vs. Unsupervised

Given: Training data: $(x_1,y_1),\ldots,(x_n,y_n)$ / $x_i \in \mathbb{R}^d$ and y_i is the label.

sample $x_1 \rightarrow$	$x_{11}\ x_{12}\ \ldots\ x_{1d}$	$y_1 \leftarrow$ label
...	... ...	...
sample $x_i \rightarrow$	$x_{i1}\ x_{i2}\ \ldots\ x_{id}$	$y_i \leftarrow$ label
...	... ...	...
sample $x_n \rightarrow$	$x_{n1}\ x_{n2}\ \ldots\ x_{nd}$	$y_n \leftarrow$ label

fruit	length	width	weight	label
fruit 1	165	38	172	Banana
fruit 2	218	39	230	Banana
fruit 3	76	80	145	Orange
fruit 4	145	35	150	Banana
fruit 5	90	88	160	Orange
...				
fruit n	...	...	...	...

Supervised vs. Unsupervised

fruit	length	width	weight	label
fruit 1	165	38	172	Banana
fruit 2	218	39	230	Banana
fruit 3	76	80	145	Orange
fruit 4	145	35	150	Banana
fruit 5	90	88	160	Orange
...				
fruit n	...	...	...	...

Supervised learning:
Learning a model from labeled data.
Unsupervised learning:
Learning a model from unlabeled data.

Supervised Learning

Training data: "samples" x with "labels" y.

$$\{(x_1, y_1), ..., (x_n, y_n)\} \ / \ x_i \in \mathbb{R}^d$$

- Classification: y is discrete. To simplify, $y \in \{-1, +1\}$

$$f : \mathbb{R}^d \to \{-1, +1\} \text{ (} f \text{ is called a binary classifier).}$$

Function f, which is a separating hyperplane is learned using data samples x with labels y to separate the two classes, such that when a new sample comes, f can predict its label to be $+1$ or -1.

Supervised Learning

Training data: "samples" x with "labels" y.

$$\{(x_1, y_1), ..., (x_n, y_n)\} \ / \ x_i \in \mathbb{R}^d$$

- Classification: y is discrete. To simplify, $y \in \{-1, +1\}$

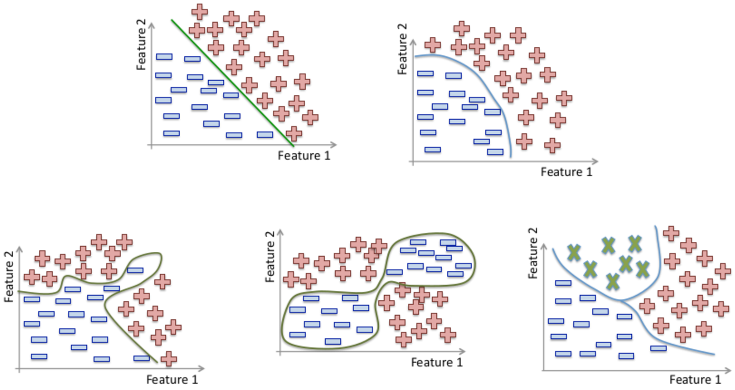
$$f : \mathbb{R}^d \rightarrow \{-1, +1\} \text{ (} f \text{ is called a binary classifier).}$$

Function f, which is a separating hyperplane is learned using data samples x with labels y to separate the two classes, such that when a new sample comes, f can predict its label to be $+1$ or -1.

Task: to predict label(s) for new (unseen) sample(s).
Solution: learn the hyperplane (f) using training data.
Evaluation: given the correct label, the predicted label(s) in the task is/are correct or not. If yes, the accuracy increases otherwise it is a misclassification.

Supervised Learning

Classification

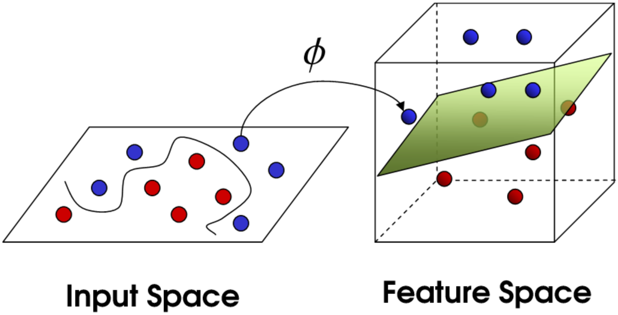


Supervised Learning

Non linear classification

If we have data $x_i, x_j \in \mathbb{R}^2$ and feature mapping, $\phi : \mathbb{R}^2 \rightarrow \mathbb{R}^3$

$$(x_{i1}, x_{i2}) \mapsto (z_{i1}, z_{i2}, z_{i3}) = (x_{i1}^2, \sqrt{2}x_{i1}x_{i2}, x_{i2}^2)$$



The data, which is not linearly separable in n dimensional space may be linearly separable in a higher dimensional space.

Supervised Learning

Training data: "samples" x with "labels" y.

$$\{(x_1, y_1), ..., (x_n, y_n)\} \ / \ x_i \in \mathbb{R}^d$$

- Regression: y is a real value. To simplify, $y \in \mathbb{R}$

$$f : \mathbb{R}^d \rightarrow \mathbb{R} \ (f \text{ is called a regressor}).$$

Function f, which is a curve is fit using data samples x with labels y, such that when a new sample comes, f can predict its label to be a real value.

Supervised Learning

Training data: "samples" x with "labels" y.

$$\{(x_1, y_1), ..., (x_n, y_n)\} \; / \; x_i \in \mathbb{R}^d$$

- Regression: y is a real value. To simplify, $y \in \mathbb{R}$

$$f : \mathbb{R}^d \rightarrow \mathbb{R} \text{ (} f \text{ is called a regressor).}$$

Function f, which is a curve is fit using data samples x with labels y, such that when a new sample comes, f can predict its label to be a real value.

Task: to predict label(s) for new sample(s).
Solution: fit the curve (f) using training data.
Evaluation: given the correct label, the predicted label(s) in the task is/are correct or not.

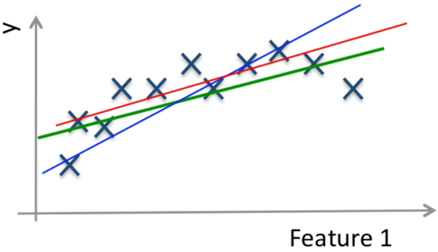
Supervised Learning

Regression

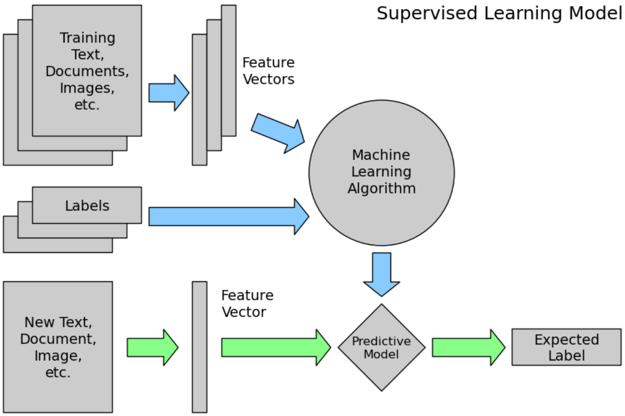
Example: Income in function of age, weight of the fruit in function of its length.

Supervised Learning

Regression



Supervised Learning Model[6]



[6]Image Source: Google

Unsupervised Learning

Training data: "samples" x.

$$\{x_1, ..., x_n\}, x_i \in X$$

- Clustering/ segmentation:

$$f : \mathbb{R}^d \rightarrow \{C_1, ...C_k\} \text{ (set of clusters).}$$

Function f is learned using data samples x to find the clusters, such that when a new sample comes, f can predict the cluster number it belongs to.

Unsupervised Learning

Training data: "samples" x.

$$\{x_1, ..., x_n\}, x_i \in X$$

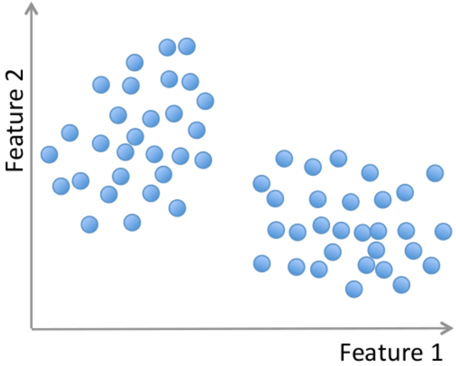
- Clustering/ segmentation:

$$f : \mathbb{R}^d \rightarrow \{C_1, ...C_k\} \text{ (set of clusters).}$$

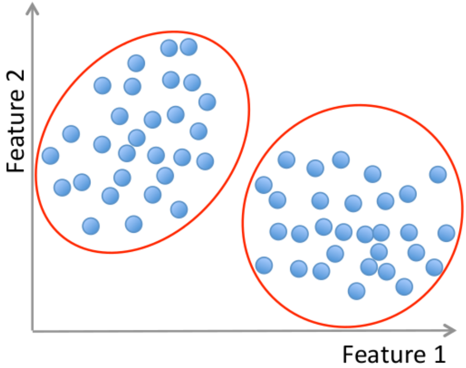
Function f is learned using data samples x to find the clusters, such that when a new sample comes, f can predict the cluster number it belongs to.

Task: to predict the cluster number(s) for new sample(s).
Solution: find the clusters (f) using training data.
Evaluation: the predicted cluster number(s) in the task is/are correct or not.

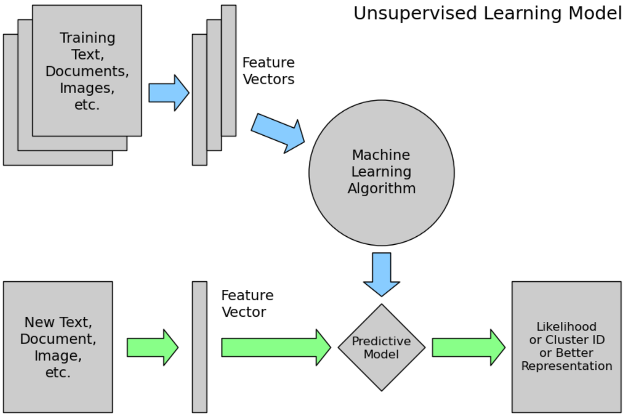
Unsupervised Learning



Unsupervised Learning

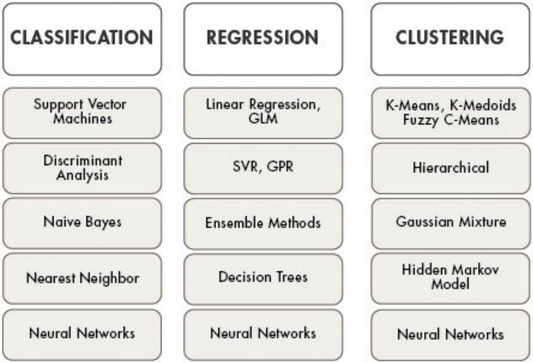


Unsupervised Learning Model[7]

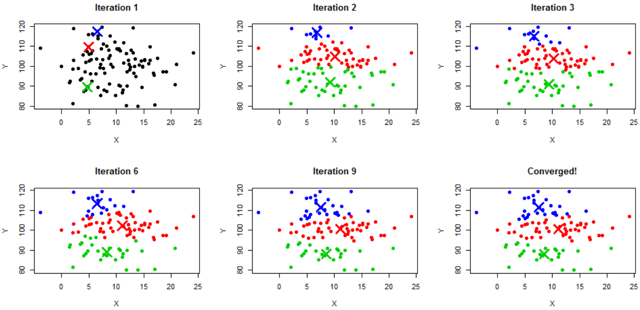


[7] Image Source: Google

Algorithms


CLASSIFICATION
REGRESSION
CLUSTERING
Support Vector Machines
Linear Regression, GLM
K-Means, K-Medoids Fuzzy C-Means
Discriminant Analysis
SVR, GPR
Hierarchical
Naive Bayes
Ensemble Methods
Gaussian Mixture
Nearest Neighbor
Decision Trees
Hidden Markov Model
Neural Networks
Neural Networks
Neural Networks

K-Means



1. Initialize **cluster centroids** $\mu_1, \mu_2, \ldots, \mu_k \in \mathbb{R}^n$ randomly.
2. Repeat until convergence: {

   For every i, set

   $$c^{(i)} := \arg\min_j \|x^{(i)} - \mu_j\|^2.$$

   For each j, set

   $$\mu_j := \frac{\sum_{i=1}^m 1\{c^{(i)} = j\} x^{(i)}}{\sum_{i=1}^m 1\{c^{(i)} = j\}}.$$

   }

Decision Trees

- Start from an empty decision tree.
- Split on next best attribute.
- Recurse.

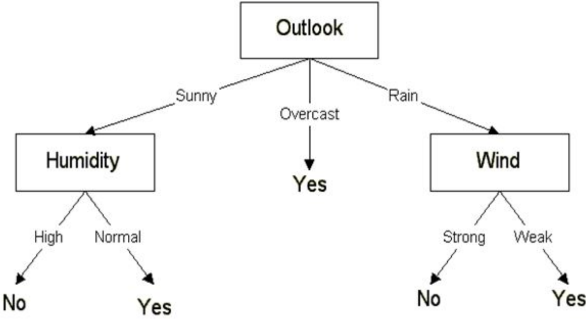


Figure: To play or not?

Problems:

- They can be extremely sensitive to small perturbations in the data: a slight change can result in a drastically different tree.
- Difficult to handle large number of features.
- Difficult to handle features with more values.
- Greedy algorithms don't necessarily yield the global optimum.
- Computationally inefficient to build large tree.

Support Vector Machines

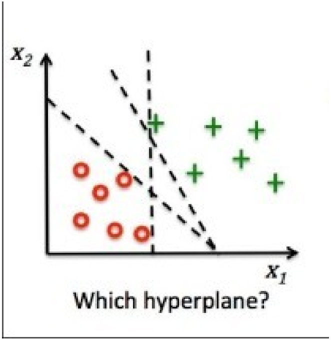


Figure: Which hyperplane is best to separate two classes[8]?

[8]Image Source: Google

Support Vector Machines

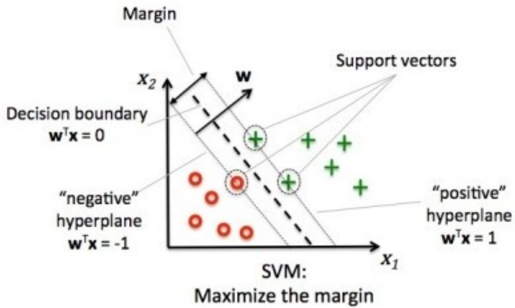


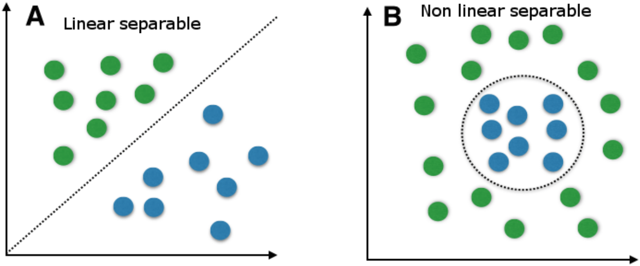
Figure: Maximum margin hyperplane[9].

Objective function:

$$\max \frac{2}{||w||} \quad s.\ t. \quad y_i(w.x_i + b) \geq 1$$

[9] Image Source: Google

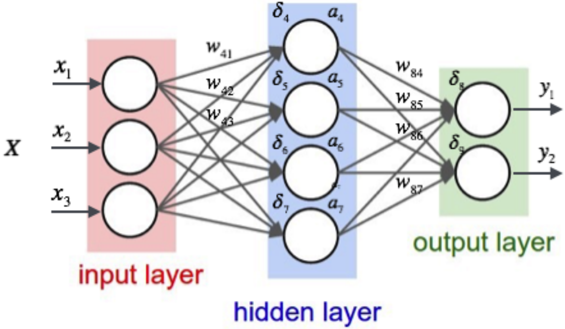
Linear vs Non-Linear Data



Problems:

- It has several key parameters that need to be set correctly to achieve the best classification results for any given problem.
- Choice of kernel for non-linear data.
- Limitation is speed and size, both in training and testing.
- The standard SVM is meant for binary classification. The optimal design for multiclass SVM classifiers is still a further area for research.

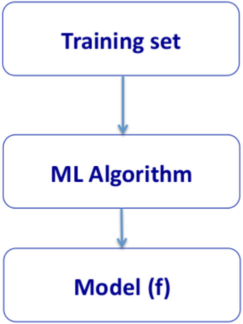
Neural Network



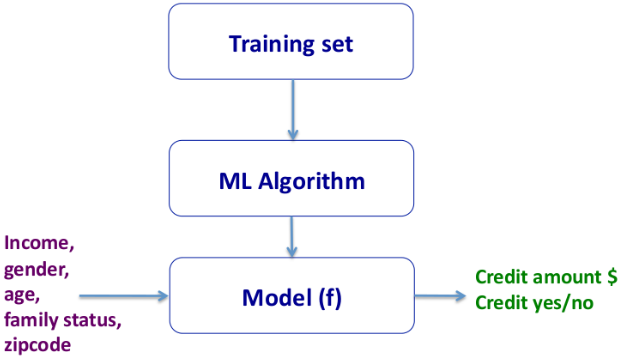
Advantages:

- NNs have the ability to learn and model non-linear and complex relationships.
- NNs can generalize: After learning from the initial inputs and their relationships, it can infer unseen relationships on unseen data as well.
- It can handle high dimensional data.

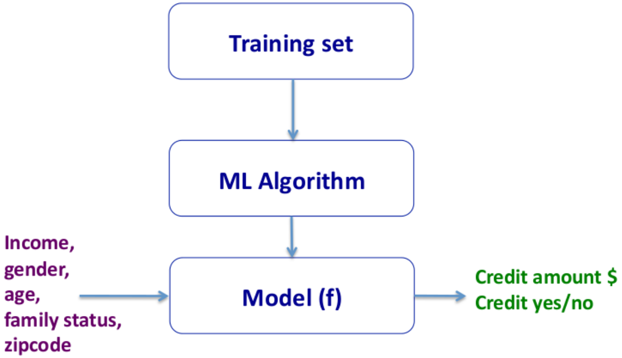
Training and Testing



Training and Testing



Training and Testing



Question: How can we be confident about f?

Training and Testing

- We calculate E^{train} the in-sample error (training error or empirical error/risk).

$$E^{train}(f) = \sum_{i=1}^{n} loss(y_i, f(x_i))$$

Training and Testing

- We calculate E^{train} the in-sample error (training error or empirical error/risk).

$$E^{train}(f) = \sum_{i=1}^{n} loss(y_i, f(x_i))$$

- Example of loss functions:
  -Classification error:

$$loss(y_i, f(x_i)) = \begin{cases} 1, & if \ sign(y_i) \neq sign(f(x_i)) \\ 0, & otherwise \end{cases}$$

Training and Testing

- We calculate E^{train} the in-sample error (training error or empirical error/risk).

$$E^{train}(f) = \sum_{i=1}^{n} loss(y_i, f(x_i))$$

- Example of loss functions:
  -Classification error:

$$loss(y_i, f(x_i)) = \begin{cases} 1, & if \ sign(y_i) \neq sign(f(x_i)) \\ 0, & otherwise \end{cases}$$

  -Least square error:

$$loss(y_i, f(x_i)) = (y_i - f(x_i))^2$$

Training and Testing

- We calculate E^{train} the in-sample error (training error or empirical error/risk).

$$E^{train}(f) = \sum_{i=1}^{n} loss(y_i, f(x_i))$$

Training and Testing

- We calculate E^{train} the in-sample error (training error or empirical error/risk).

$$E^{train}(f) = \sum_{i=1}^{n} loss(y_i, f(x_i))$$

- We aim to have $E^{train}(f)$ small, i.e., minimize $E^{train}(f)$

Training and Testing

- We calculate E^{train} the in-sample error (training error or empirical error/risk).

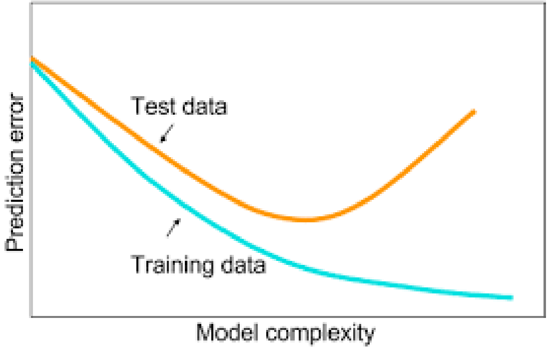
$$E^{train}(f) = \sum_{i=1}^{n} loss(y_i, f(x_i))$$

- We aim to have $E^{train}(f)$ small, i.e., minimize $E^{train}(f)$
- We hope that $E^{test}(f)$, the out-sample error (test/true error), will be small too.

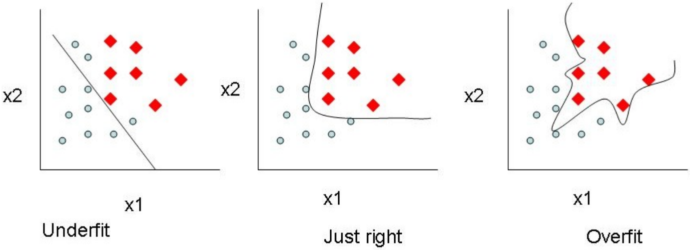
Outline

Generalization?

Generalization refers to your model's ability to adapt properly to new, previously unseen data, drawn from the same distribution as the one used to create the model.



Problem with Classification



- Overfitting refers to a model that models the training data too well.
- Overfitting happens when a model learns the detail and noise in the training data to the extent that it negatively impacts the performance of the model on new data.
- Underfitting refers to a model that can neither model the training data nor generalize to new data.
- Good fit is to select a model at the sweet spot between underfitting and overfitting.

Problem with Regression

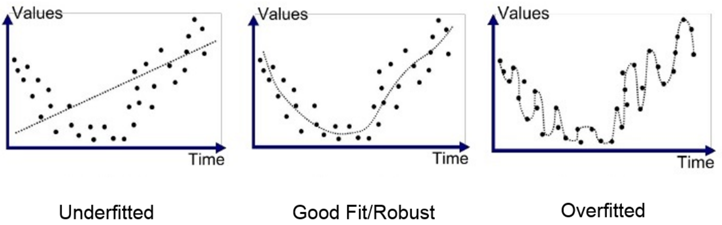


Figure: Generalization problem with regression[10].

$$f(x) = \lambda_0 + \lambda_1 x \ \dots \ (1)$$

$$f(x) = \lambda_0 + \lambda_1 x + \lambda_2 x^2 \ \dots \ (2)$$

$$f(x) = \lambda_0 + \lambda_1 x + \lambda_2 x^2 + \lambda_3 x^3 + \lambda_4 x^4 \ \dots \ (3)$$

[10] Image Source: Google

Problem with Clustering

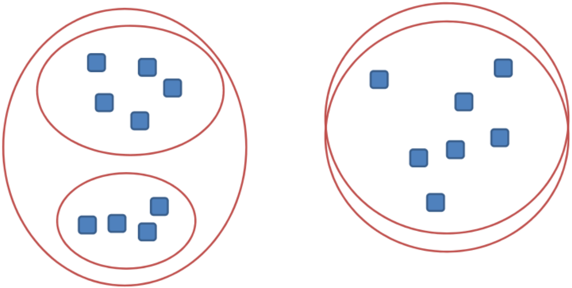


Figure: Generalization problem with clustering[11].

[11] Image Source: Google

Avoid Overfitting

In general, use simple models!

- **Reduce** the number of features manually or do feature selection.
- Do a **model selection**.
- Use **regularization** (keep the features but reduce their importance by setting small parameter values).
- Do a **cross-validation** to estimate the test error.

Regularization

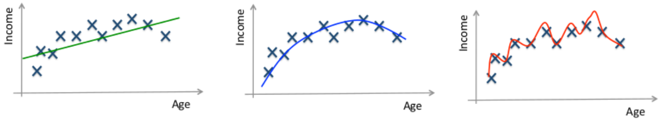
We want to minimize:

Classification term + λ x Regularization term

$$\sum_{i=1}^{n} loss(y_i, f(x_i)) + \lambda \times R(f)$$

This is a form of regression, that constrains/ regularizes or shrinks the coefficient estimates towards zero. In other words, this technique discourages learning a more complex or flexible model, so as to avoid the risk of overfitting.

Regularization



$$f(x) = \lambda_0 + \lambda_1 x \ \dots \ (1)$$

$$f(x) = \lambda_0 + \lambda_1 x + \lambda_2 x^2 \ \dots \ (2)$$

$$f(x) = \lambda_0 + \lambda_1 x + \lambda_2 x^2 + \lambda_3 x^3 + \lambda_4 x^4 \ \dots \ (3)$$

Hint: Avoid high-degree polynomials.

If there is noise in the training data, then the estimated coefficients won't generalize well to the future data. This is where regularization comes in and shrinks or regularizes these learned estimates towards zero.

Train, Validation and Test

Example: Split the data randomly into 60% for training, 20% for validation and 20% for testing.

Train, Validation and Test

1. Training set is a set of examples used for learning a model (e.g., a classification model).

Train, Validation and Test

1. Training set is a set of examples used for learning a model (e.g., a classification model).
2. Validation set is a set of examples that cannot be used for learning the model but can help tune model parameters (e.g., selecting K in K-NN). Validation helps control overfitting.

Train, Validation and Test

1. Training set is a set of examples used for learning a model (e.g., a classification model).
2. Validation set is a set of examples that cannot be used for learning the model but can help tune model parameters (e.g., selecting K in K-NN). Validation helps control overfitting.
3. Test set is used to assess the performance of the final model and provide an estimation of the test error.

k-fold Cross Validation

A method for estimating test error using training data.

Algorithm:

Given a learning algorithm $\mathcal{A}$ and a dataset $\mathcal{D}$

Step 1: Randomly partition $\mathcal{D}$ into k equal-size subsets $\mathcal{D}_1, \dots, \mathcal{D}_k$.

Step 2: Train $\mathcal{A}$ for each subset of $\mathcal{D}$.

Step 3: Average error over all folds.

$$\sum_{j=1}^{k} (E^{\mathcal{D}_j})$$

Outline

Evaluation Metrics

Confusion Matrix

		Actual Label	
		Positive	Negative
Predicted Label	Positive	True Positive (TP)	False Positive (FP)
	Negative	False Negative (FN)	True Negative (TN)

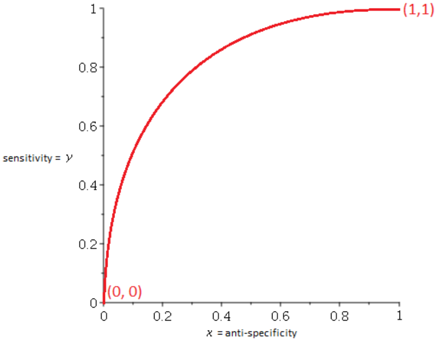
Evaluation Metrics

Confusion Matrix

		Actual Label	
		Positive	Negative
Predicted Label	Positive	True Positive (TP)	False Positive (FP)
	Negative	False Negative (FN)	True Negative (TN)

Accuracy	(TP + TN) / (TP + TN + FP + FN)	The percentage of predictions that are correct
Precision	TP / (TP + FP)	The percentage of positive predictions that are correct
Sensitivity (Recall)	TP / (TP + FN)	The percentage of positive cases that were predicted as positive
Specificity	TN / (TN + FP)	The percentage of negative cases that were predicted as negative

Receiver Operating Characteristic Curve



ROC is a probability curve and AUC represents degree or measure of separability. It tells how much model is capable of distinguishing between classes.


Questions