100 Page Machine Learnings

Books Outline

Types of learning
Supervised learning

Notation
Randomness
unbiased estimations
Baye's rule
parameters
classification vs. regression
shallow vs. deep learning

Linear
Logistic
Decision trees
SVMs
K- nearest neighbours
Learning algorithms
Basic practices
Neural networks and deep learning
problems & solutions
Advanced practice
unsupervised learning

Machines don't learn

Artificial intelligence is not intelligence

Introduction

Gather dataset
Build Statistical model on that dataset.

Supervised learning: labelled examples, of feature vectors x_i

$\{(x_i, y_i)\}_{i=1}^{N}$

Each dimension $1, 2 \ldots j \ldots D$ describes the example

Feature: $x^{(j)}$

Feature positions contain the same kind of information for uniformity

y_i belongs to a set of classes, real #, vector, matrice, tree, or graph

Goal of supervised learning: Model takes in feature vector x as input and outputs a label for the feature vector.

Unsupervised learning: $\{x_i\}_{i=1}^{N}$

Transforms vector into another vector that can be used to solve a problem

Semi-supervised learning
Both labelled and unlabelled

more data = better since a larger sample better represents the probability distribution the data came from.

Reinforcement learning:

The machine "lives" in an environment and is capable of perceiving the state of the environment as a vector of features

The machine can execute actions in that state

Different actions → different rewards / move machine to another state of the environment

The goal of RL models is to learn policy
Takes a feature vector and outputs the optimal/desired action to execute in that state

=> For sequential and long-term decisions

How supervised learning works:

(input, output) pairs
⇓ ⇓
Machine-readable data

SVMs: Every input is a point in a high dimensional space

Imaginary n-1 hyperplane, and draws a line that separates examples of different classes → decision boundary

Equation of the hyperplane: $wx - b = 0$

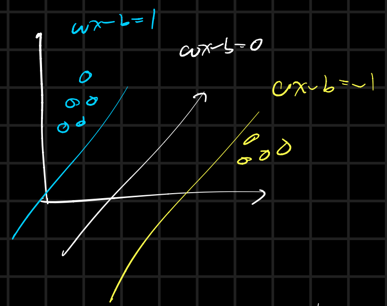
w = real-valued vector

$\overline{wx} = w^1 x^1 + w^2 x^2 + \dots w^D x^D$, for D = dim

predicted label: $y = \text{sign}(wx - b)$

Find optimal values for w and b

we want model to predict the input labels correct



want separation with large margin.

Distance between closest 2 examples defined by decision boundary ($wx-b=0$ line)

=> minimize Euclidean norm of w ($\|w\|$) $\sqrt{\sum_{j=1}^{D}(w^{(j)})^2}$

minimize $\|w\|$ subject to $y_i(wx_i - b) \geq 1$ for $i = 1, \ldots N$

Distance between the 2 planes: $\frac{2}{\|w\|}$

as $\|w\| \downarrow$, distance increases

Any classification model creates a decision boundary

Accuracy vs. speed?

The decision boundary should separate classes, given the data of

a class is chosen independently of other data chosen.

May not be effective for outliers.

The larger a dataset used, the less likely unseen examples lie on the plot far from examples

used for training

SVM decision boundary is as far as possible from examples of both classes.