# Nearest Neighbour Algorithms

Christos Dimitrakakis

September 20, 2024

# Outline

## Introduction

The hidden secret of machine learning

## The algorithm

$k$ Nearest Neighbours

Extensions and parameters

## Activities

## Introduction

The hidden secret of machine learning

## The algorithm

*k* Nearest Neighbours

Extensions and parameters

## Activities

# Supervised learning

- Given labelled training examples $(x_1, y_1), \dots (x_T, y_T)$ where
- $x_t \in X$, the input variables (or features)
- $y_t \in Y$, the output variables (or labels)

## Feature space $\mathcal{X}$

- Usually $\mathcal{X} = \mathbb{R}^n$: the n-dimensional Euclidean space
- How do we use your class data?

## Classification

- $Y = \{1, \dots, m\}$ are discrete labels

## Regression

- $Y = \mathbb{R}^m$ are continuous values

# The kNN algorithm idea

- Assume an unknown example is similar to its neighbours
- Smoothness allows us to make predictions

*Discriminatory analysis-nonparametric discrimination: consistency properties*, Evelyn Fix and Joseph L. Hodges Jr, 1951.

Figure: Evelyn Fix

Figure: Joseph Hodges

# Performance of KNN on image classification

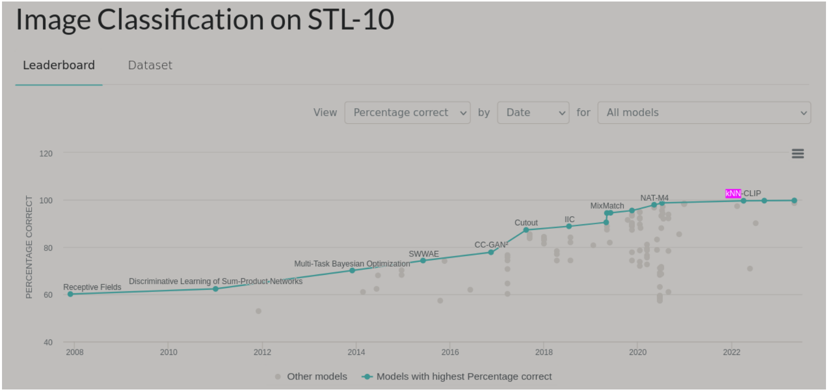

Image Classification on STL-10

View Percentage correct by Date for All models

- Really simple!
- Can outperform really complex models!

Introduction

- The hidden secret of machine learning

The algorithm

- $k$ Nearest Neighbours
- Extensions and parameters

Activities

# The Nearest Neighbour algorithm

## Pseudocode

```
Input: Data $(x_t, y_t)_{t=1}^{T}$, test point $x$, distance $d$
$t^* = \arg\min_t d(x_t, x)$ / How do we implement this?
Return $\hat{y}_t = y_{t^*}$
```

## Classification

$\hat{y}_t \in [m] \equiv \{1, \ldots, m\}$

## Regression

$\hat{y}_t \in \mathbb{R}^m$

# The k-Nearest Neighbour algorithm

## Pseudocode

- Input: Data $(x_t, y_t)_{t=1}^T$, test point $x$, distance $d$, neighbours $k$
- Calculate $h_t = d(x_t, x)$ for all $t$.
- Get sorted indices $s = \texttt{argsort}(h)$ so that $d(x_{s_i}, x) \leq d(x_{s_{i+1}}, x)$ for all $i$. (How?)
- Return $\sum_{i=1}^k y_{s_i}/k$.

## Classification

- It is not convenient to work with discrete labels.
- We use a one-hot encoding $(0, \ldots, 0, 1, 0, \ldots, 0)$.
- $y_t \in \{0, 1\}^m$ with $\|y_t\|_1 = 1$, so that the class of the $t$-th example is $j$ iff $y_{t,j} = 1$.

## Regression

- $y_t \in \mathbb{R}^m$, so we need do nothing

# Making a decision

## kNN: A model of the conditional distribution $P(y|x)$

- Given features $x$, we get a vector
- $p_i = \hat{\mathbb{P}}(y = i|x)$.

## The optimal decision rule $\pi$ derived from kNN

- Classification decision $a_t \sim \pi(a|x_t)$
- $a_t \in$ but $\neq \mathcal{Y}$, e.g. can include "Do not Know", or "Alert" etc.
- Actual label $y_t$
- $U(a_t, y_t)$: utility function depending on the application.

## Decision rule maximising accuracy

- $a_t = \arg\max_i \hat{\mathbb{P}}(y = i|x)$.

# The number of neighbours

$k = 1$

- How does it perform on the training data?
- How might it perform on unseen data?

$k = T$

- How does it perform on the training data?
- How might it perform on unseen data?

# Distance function

## For data in $\mathbb{R}^n$, $p$-norm

$$d(x, y) = \|x - y\|_p$$

## Scaled norms

When features having varying scales:

$$d(x, y) = \|Sx - Sy\|_p$$

Or pre-scale the data

## Complex data

- Manifold distances
- Graph distance

# Distances

## A distance $d(\cdot, \cdot)$:

- Identity $d(x, x) = 0$.
- Positivity $d(x, y) > 0$ if $x \neq y$.
- Symmetry $d(y, x) = d(x, y)$.
- Triangle inequality $d(x, y) \leq d(x, z) + d(z, y)$.

## For data in $\mathbb{R}^n$, $p$-norm

$$d(x, y) = \|x - y\|_p$$

# Norms;

## A norm $\| \cdot \|$

- Zero element $\|0\| = 0$.
- Homogeneity $\|cx\| = c\|x\|$ for any scalar $a$.
- Triangle inequality $\|x + y\| \leq \|x\| + \|y\|$.

## \$p\$-norm

$$\|z\|_p = \left( \sum_i z_i^p \right)^{1/p}$$

# Neighbourhood calculation

If we have $T$ datapoints

## Sort and top $K$.

- Requires $O(T \ln T)$ time

## Use the Cover-Tree or KD-Tree algorithm

- Requires $O(cK \ln T)$ time.
- $c$ depends on the data distribution.

Introduction

The hidden secret of machine learning

The algorithm

*k* Nearest Neighbours

Extensions and parameters

Activities

# KNN activity

- Implement nearest neighbours
- Introduction to scikitlearn nearest neighbours
- Introduction to generalisation errors