Outline

% Created 2023-09-26 Di 08:17 % Intended LaTeX compiler: pdflatex [presentation]beamer [utf8]inputenc [T1]fontenc graphicx grffile longtable wrapfig rotating [normalem]ulem amsmath textcomp amssymb capt-of hyperref {{{$\mathbb{E}$}}} {{{$\mathbb{I}$}}} \#1\} {{{{$\mathbb{P}$}}}} $*$ *arg max* $*$ *arg min* $\triangleq$}$\mathbb{R}$} $\Pr\Theta$} $\Pr\theta$} []beamerthemedefault { pdfauthor={Christos Dimitrakakis}, pdftitle={Nearest Neighbour Algorithms}, pdfkeywords={}, pdfsubject={}, pdfcreator={Emacs 26.3 (Org mode 9.1.9)}, pdflang={English}}

Nearest Neighbour Algorithms

Christos Dimitrakakis

October 10, 2023

Outline

Supervised learning

- Given labelled training examples $\Pr(x_1, y_1), \ldots (x_T, y_T)$ where
- $\Pr x_t \in^n$ are features
- $\Pr y_t \in Y$ are labels

Classification

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

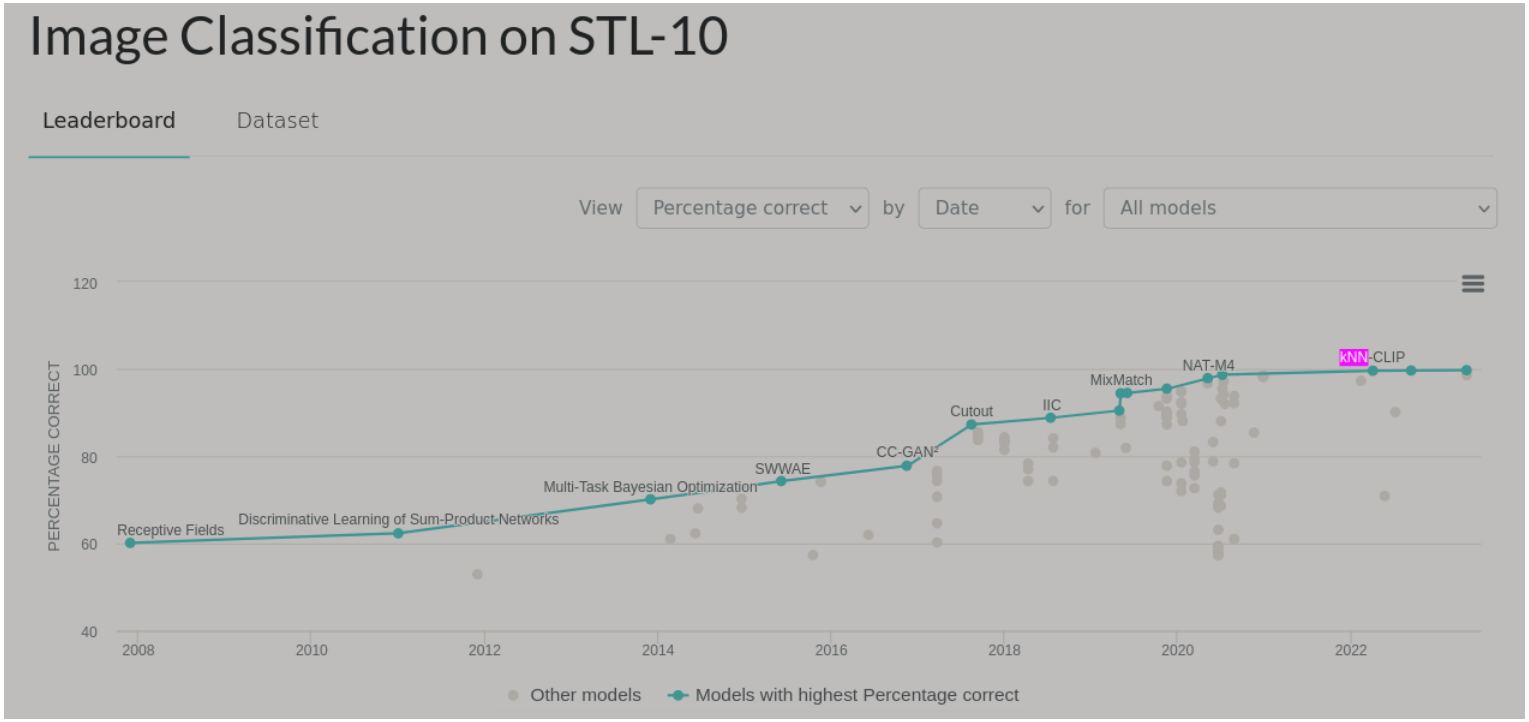
Regression

- $\Pr Y =^m$ are continuous values

The kNN algorithm idea

- Assume an unknown example is similar to its neighbours
- Smoothness allows us to make predictions
- Lots of other algortihms

Performance of KNN on image classification



- Really simple!
- Can outperform really complex models!

The Nearest Neighbour algorithm

Pseudocode

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

Classification

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

Regression

$\Pr \hat{y}_t \in^m$

The k-Nearest Neighbour algorithm

Pseudocode

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

Classification

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

Regression

- $\Pr y_t \in^m$, so we need do nothing

The number of neighbours

$\Pr k = 1$

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

$\Pr k = T$

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

Distance function

For data in Pr^n, Pr 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 $\Pr d(\cdot, \cdot)$:

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

For data in $\Pr^n$, \$p\$-norm

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

Norms;

A norm Pr $\|\cdot\|$

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

p-norm

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

Neighbourhood calculation

If we have $\Pr T$ datapoints

Sort and top $\Pr K$.

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

Use the Cover-Tree or KD-Tree algorithm

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

Class data

Fill in the class data

Figure: Link to spreadsheet

KNN activity

- Implement nearest neighbours
- Introduction to scikitlearn nearest neighbours