Anatomy of ML

1. Dataset (D)
   - ↳ Feature Vector ($\vec{x}$)
   - ↳ output (y)
2. Algorithm (h)
   - $\mathcal{H}$ denotes all possible algorithm Learning: picks best $h \in \mathcal{H}$
   - $h \in \mathcal{H}$
   - $h(\vec{x}) = y$
3. Loss function
   - $\ell(h, D)$
   - Want to minimize values of ℓ

But we don't want to minimize in-sample loss.
Minimize on new data

$\mathbb{E}[\ell(h, (\hat{x}, y))]_{(x,y)\sim P}$ However, distribution of P unknown.

Approximate $\mathbb{E}[\ell(h, (\hat{x}, y))]$ with the test set

$$\ell(h, D_{TR}) = \frac{1}{n}\sum_{i=1}^{n} \ell(h, (x_i, y_i))$$

By WLLN, $\lim_{n\to\infty} \frac{1}{n}\sum_{i=1}^{n} \ell(h, (x_i, y_i)) = \mathbb{E}[\ell(h, (\hat{x}, y))]$

(for non-pathological fn)

Perform test/train set with care

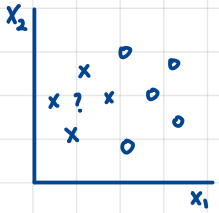
- Spam: Temporal split better than random split
  - Spammers send email many times. Emails in test may also be in train

If you have many models

- Train / Validation / Test split
- Pick your champion model using validation set
- Report performance metric w/ test set

K-Nearest Neighbor (1967)

Binary Classification Problem



Assumption: Data pts that are closest are similar

Test point x

Let $S_x \subseteq D$ s.t. $|S_x| = k$

$\forall (x', y') \in D \setminus S_x \quad \text{dist}(x, x') \geq \max_{x'' \in S_x} \text{dist}(x, x'')$

} i.e. S_x is the set of k nearest neighbor of test pt x.

Within S_x, the most prevalent label is the classification result

Minkowski Metric

$$\text{dist}(x, z) = \sum_{r=1}^{d} \left(|[x]_r - [z]_r|^p \right)^{\frac{1}{p}}$$

r_{th} dimension of x, z

$p = 1$	Manhattan dist
$p = 2$	Euclidian dist
$p \to \infty$	Max

$32^{1000} + 15^{1000} \simeq 32^{1000}$

How to choose k?

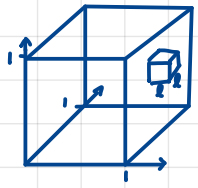
Elbow Method. Lowest k that minimizes loss

What happens if you ↑ k?

The more prevalent label gets a boost

Curse of Dimensionality

K-Nearest neighbor **NOT suitable for high dimensional data**



n pts randomly distributed in hypercube w/ length 1 in dim d

How large does the smaller hypercube have to be to contain k-nearest neighbor?

$$\frac{l^d}{1^d} \approx \frac{k}{n}$$

$$l \approx \left(\frac{k}{n}\right)^{\frac{1}{d}}$$

Say $\frac{k}{n} = \frac{1}{100}$

d	l
2	0.1
10	0.63
100	0.95

If you look at knn, and contain them in a box the box is almost as big as the distr Space itself.

The k-nearest neighbor are not near at all

All pts roughly the same distance from each other

Interior is empty

Here is another way to look at it

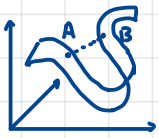
If one of the coordinate is on the edge, the pt is on the edge

$$P\{\text{interior}\} = (1-2\varepsilon)^d \xrightarrow{d\to\infty} 0$$

To apply KNN, data must have low intrinsic dimension

1. Data lies on a subspace in high dim space

2. Data lies on a low dim manifold



Locally Euclidian

A, B far on manifold distance

But close on euclidian distance

Would be a problem except we are only looking at neighbors

Advantages & Disadvantages of KNN

Pros

- Good classifier if $n \to \infty$

Cons

- Curse of dimensionality
- Long Runtime $O(nd)$
- Can't deal w/ non linear bdry.