Intro to Machine Learning and Binary Classification

An aggregation of class notes and Lugosi notes

Aimee Barciauskas

2016-02-08

Nearest Neighbor

Suppose that $\mathcal{X}$ is a metric space, distance is a function of $X \in \mathcal{X}$.

Properties of distances:

- $d(x, z) > 0 \rightarrow x \neq z$
- $d(x, x) = 0$
- $d(x, z) = d(z, x)$
- $d(x, z) \leq d(x, v) + d(v, z)$ the "triangle inquality"

Examples of how to measure metrix space: L_2 (Euclidian, proof of the triangle inquality), L_1 ("Manhattan distance"), L_{inf}

1-Nearest Neighbor

Algorithm:

1. Order distances: $d(X_{(1)}, x) \leq d(X_{(2)})$
2. Label by nearest point: $g_n(x) = Y_{(1)}(x)$

Proof of proximity

Why can we assume the nearest neighbor is close? **EXERCISE**

Answers:

- Problemset 1, #5
- Nearest Neighbor Notes, page 3

Properties of Nearest Neighbors

- **curse of dimensionality:** As d grows, required points grows exponentially
- The nearest neighbor is *typically* at a distance at most $cn^{-\frac{1}{2}}$ where c is a constant. Here *typically* indicates that X has a positive and smooth density.

Theorem: the risk of the NN classifier is such that, for all distributions of (X, Y):

$$R(g_n) \rightarrow 2\mathbb{E}[\eta(x)(1 - \eta(x))]$$

and

$$R^* \leq R^{NN} \leq 2R^*$$

In particular if $R^* = 0$ then $R^{NN} = 0$

K-Nearest Neighbor

Let $k \geq 1$ be an odd integer.

Let

$$g_n(x) = \begin{cases} 1 \text{ if } \sum Y_i(x) > \frac{k}{2} \\ 0 \text{ otherwise} \end{cases}$$

Why bother?

Suppose $Y \sim Bern(0.9)$ and $Y' \sim Bern(0.9)$. Then, 1-NN risk is $2p(1-p)$.

If $Y' = (Y'_1, ..., Y'_k)$, risk is $\mathbb{E}\{\mathbb{P}[majority((Y'_1, ..., Y'_k)) \neq Y | X]\}$

For example, for 3-NN:

$$\mathbb{P}(majority((Y'_1, Y'_2, Y'_3)) \neq Y) = \eta(1-\eta)^3 + 3\eta^2(1-\eta)^2 + \eta^3(1-\eta) + 3\eta^2(1-\eta)^2$$

e.g. first term is probability all of $Y' \neq Y$, second term equivalent to probability 2 of $Y' \neq Y$

This can be reduced to: $R^{3-NN} = \mathbb{E}[\eta(x)(1-\eta(x))] + 4\mathbb{E}[\eta(x)^2(1-\eta(x))^2]$ which "is peanuts"

R^{k-NN} in the general case: **EXERCISE**

Nearest Neighbor Notes, page 8

Corollaries

$$2R^* \geq R^{k-NN} \geq R^*$$

This is all distribution free! :) But it is also asymptotic :(

K as a function of n

One can take $k = k(n)$

Still we have

$$\|X_{k(n)} - x\| \to 0 \text{ as } n \to \infty$$

if $\frac{k(n)}{n} \to 0$

Moreover, if $k(n) \to \infty$ and $\frac{k(n)}{n} \to 0$, then, for all distributions:

$$R(g_n^{k(n)}) \to R^* \text{ in probability}$$

- (LLN?) $k(n) \to \infty$ is important for averaging to work, the variance is small.
- $\frac{k(n)}{n} \to 0$ implies bias is small

Thus, the $k(n) - NN$ rule is univisally consistent

> The real challenge in machine learning is to focus on where the action is. (Gabor)

Table 1:

d (dimension)	h (length of one side of hypercube)
1	0.01
2	0.1
10	0.63 (how is this local?)
100	0.955
1000	0.9954

Other local averaging rules

1. Histogram Rules

Split the feature space into "small" cells and take the majority vote of points in the same cell. In $\mathbb{R}^d$ one may partitiaon the space into cubes of length h. Same tradeoffs fro h as k in $k - nn$ * **One can prove that if $h(n) \to 0$ and $n(h(n)^d) \to \infty$ where $n(h(n)^d)$ is the volume of the hyper cell.

Curse of Dimensionality

Suppose X is uniformly distributed $[0,1]^d$, choose h such that each box contains 1% of the data on average.

$$0.01 = volume(cube) = h^d \text{ entire space has volume 1}$$

2. Kernel / Nataraya-Wabon / Parzen Classifiers

Given $K : \mathbb{R}^d \to \mathbb{R}$ (typically positive and decreasing away from 0)

$$g_n(x) = \begin{cases} 1 & \text{if } \sum_{i:Y_i=1} K\frac{x-X_i}{h(n)} > \sum_{i:Y_i=0} K\frac{x-X_i}{h(n)} \\ 0 \text{ otherwise} \end{cases}$$

The first term acts as a weight, while the second term makes it local.