Nearest Neighbors

Nishanth Nair

10/6/2019

Overview

- Also called instance-based or nonparametric modeling.
- There is no training step. Training is loading all the data in memory.
- Predicts the same label as the nearest neighbor in the training data.
- **K-Nearest Neighbors(k-NN)**:
  - Take a vote of k nearest neighbors.
- **Weighted K-Nearest Neighbors**:
  - Weight each label by its similiraty (inverse distance)
  - No need to choose a value of k,since points far away become less important.But this approach increses computation cost.
  - Unweighted voting will be biased by non-uniform distribution of training examples.So weighted k-nn is preffered.
- **Edited K-Nearest Neighbors**:
  - Remove outliers if all neighbors are in a different class
  - Reduces impact of outliers in the training set maybe caused due to mislabeled examples or unlearnable examples.
- K-NN can be used for classification as well as regression:
  - In classification - Asssign the class of the nearest neighbor(s)
  - In regression - Assign the average value of neighbor(s).

Distance Metrics (who is the nearest neighbor)

For numeric features

Metric 1: $L^n(x_1, x_2) = \sqrt[n]{\sum_i^N |x_{1,i} - x_{2,i}|^n}$

where,

- x_1, x_2 are vectors
- Euclidean distance, n=2
- Manhattan distance, n=1

- $\lim_{x \to \infty}$, L^n is the max distance among all components of x.
- L^n Norm

Metric 2: Cosine Similarity

- Only angle between vectors matter
- Similarity is the size of the angle

For two vectors A and B,

Cosine similarity = $\frac{A.B}{||A||.||B||} = \frac{\sum_{i=1}^{n} A_i X B}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \cdot \sqrt{\sum_{i=1}^{n} (B_i)^2}}$

For symbolic features

These could be features that have binay values categorical values.

Metric 1: Hamming distance - Measure of overlap between 2 vectors.

Example:

A - 1011001001

B - 1001000011

Overlap = Hamming distance = 3

Curse of dimensionality

- Adding dimensions increases feature space exponentially.
- Multidimensional data is difficult for K-NN

Options to deal with curse of dimensionality for k-NN

- Normalize numeric values (mean=0, variance=1)
- Feature selection: Choose subset
- Forward selection: Choose single best feature first
- Backward selection: Remove least important feature first.
- Avoid k-NN

Voronoi Diagrams

- Shows regions in space closest to each point. This what K-NN tries to learn.
- K-NN has complex boundaries

- As k increases, boundary starts to smooth. Better generalizes to new data.

Ockham's Razor

- Always opt for the expanation with the fewest possible factors.
- Consistency vs simplicity
- Parsimony: Explanation of the data should be simple
- Add complexity only if necessary

Efficiency Techniques

- **Prototypes**: Remove training examples if all neighbors are in the same class.This essentially means they donot contribute to the decision boundary and thus removing them won't impact the results. This reduces data size making inference faster.
- **Coarse to fine**: Choose simplified features to choose candidate neighbors.Again, reduce data size.Then use full feature sset to perform final distance measurement.
- **Neighbor graph**: Since all points are not relevant for inference, create a neighbor graph using a KD-Tree (Similar to binary search) to partition feature space.Then use this to get a filtered dataset of points near our target to run inference on.