ECE 20875
Python for Data Science
Milind Kulkarni and Chris Brinton

classification: k-nearest neighbor
Naïve Bayes is a nice classifier, but it is parametric.

We must have a model of the data in mind, and some prior knowledge, to use it effectively.

What if we don’t have any such knowledge? What if all we have is our input data, and it does not seem to fit any existing distribution well?

k-nearest neighbor is a classifier that requires no assumptions about the data:

- Look at the classes of the $k$-nearest points and pick the most frequent one.
k-nearest neighbor

- Start with labeled training data, just like naïve Bayes
k-nearest neighbor

• Take new data point
k-nearest neighbor

- Draw a circle around it
k-nearest neighbor

- Grow the circle until it has \( k \) other points in it

- \( k \) is a parameter you set (e.g., 3)
k-nearest neighbor

• Grow the circle until it has $k$ other points in it

• $k$ is a parameter you set (e.g., 3)
k-nearest neighbor

- Count how many points from **class 1** are in the circle and how many from **class 2**

- **Majority wins**
k-nearest neighbor

• Count how many points from **class 1** are in the circle and how many from **class 2**
  • **Majority wins**

• How to choose $k$?
  • Larger $k$ means we are less sensitive to outliers, but also less discriminative
  • **Cross validation!**
**k-nearest neighbor**

- **Algorithmic interpretation:**
  - Find the distance $|x - x_0|$ from new point $x_0$ to every other point $x$.
  - Sort by distance, pick closest $k$ points.
  - Predicted class is the one with the most “votes” from these $k$.

- **In Python**
  - `from sklearn.neighbors import KNeighborsClassifier`

pros vs cons

+ Simple concept for classifier
+ No models or prior knowledge required

- Sensitive to outliers (new point may be close to outliers from class 0, but still belongs in class 1)

- Expensive to use model (compute distances from all other points)

- Does not help with missing data (classifier is only as good as labeled training data)