Lecture 11:
k-Nearest Neighbors (kNN) Algorithm
STAT 452, Spring 2021

kNN Algorithm

- k-Nearest Neighbors (kNN) is a simple algorithm that makes predictions for new values of the response based on similarity to observations in the training set.
- kNN can be used for both regression and classification tasks.
- In this lecture, we only focus on using kNN for classification.
- Unlike logistic regression, kNN is well-suited for response variables that have more the 2 categories (e.g., low, medium, high).

- For classification, the kNN algorithm identifies the k training set observations that are most "similar" or nearest to the new, test set observation.
- The most common class of those k training set observations (i.e., the majority vote) is the kNN prediction for the test set observation.
- The method is best explained through an illustration (next slide).

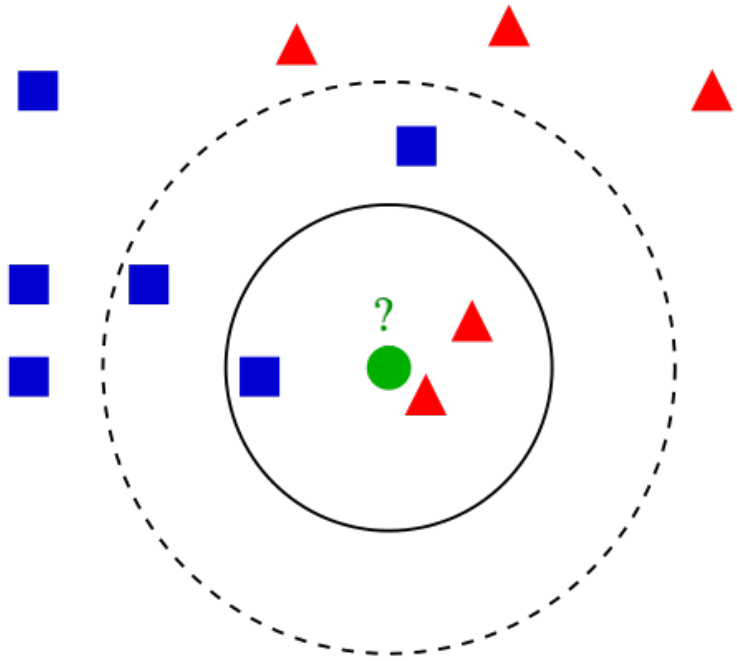

Figure: Example of k-NN classification. The test sample (green dot) should be classified either to blue squares or to red triangles. If k = 3 (solid line circle) it is assigned to the red triangles because there are 2 triangles and only 1 square inside the inner circle. If k = 5 (dashed line circle) it is assigned to the blue squares (3 squares vs. 2 triangles inside the outer circle).

Source: `https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm`

Measuring Distance

- Locating the nearest neighbors required a distance function, or a formula that measures the similarity between two observations.
- Traditionally, the kNN algorithm uses **Euclidean distance**, which is the straight line distance between two points.
- In two dimensions, the Euclidean distance between points $\boldsymbol{p} = (p_1, p_2)$ and $\boldsymbol{q} = (q_1, q_2)$ is be computed as

$$dist(\boldsymbol{p}, \boldsymbol{q}) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2}$$

- In n dimensions, the distance formula generalizes to

$$dist(\boldsymbol{p}, \boldsymbol{q}) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \cdots + (q_n - p_n)^2}$$

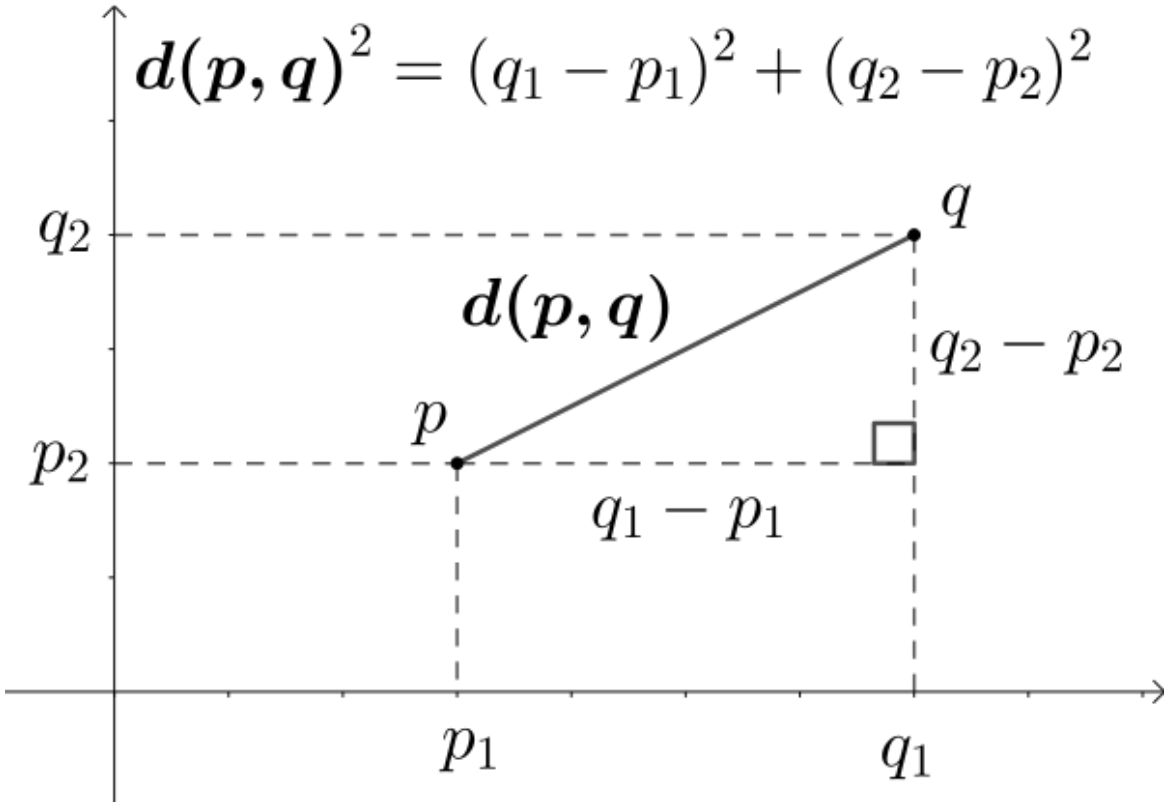

Figure: Using the Pythagorean theorem to compute two-dimensional Euclidean distance

Source: `https://en.wikipedia.org/wiki/Euclidean_distance`

Stadardization

- Predictors variables are typically standardized prior to applying the kNN algorithm.
- The reason for this is that the distance formula is affected by the scaling of the variables.
- When applying kNN, the distance measures will be dominated by variables that are on a larger scale. Standardizing the predictors is a good way to handle this issue.

Stadardization

- **Min-max standardization** transforms a variable so the values are between 0 and 1:

$$\frac{x - min(x)}{max(x) - min(x)}$$

- **Z-score standardization** transforms a variable so that the mean is 0 and standard deviation is 1:

$$\frac{x - \bar{x}}{s_x}$$

Choosing k

- When $k = 1$, then the single nearest neighbor is used for classification.
- If k is very large (say, $k \approx n$), then nearly every training observation is represented in the final vote, and so the algorithm would always predict the majority class.
- Usually, the best value of k is somewhere in between.
- Cross-validation can be used to select k. Try a variety of values for k and then select the value that gives the highest accuracy on the test set.