KNN and Trees

Josh Bernhard
Galvanize

Non-parametric vs. Parametric

Parametric
- Multiple Linear Regression
- Lasso/Ridge
- General Additive Models
- Generalized Linear Models

Non-Parametric
- KNN
- Bagging
- Boosting
- SVMs with Non-linear Kernals

No one Model beats all other models across all datasets.

Non-parametric vs. Parametric

Parametric

- Force functional form (less flexible)
- More interpretable than non-parametric methods
- If the functional form is correct – these can be quite nice, but are frequently outperformed by more flexible models
- Simple to perform statistical tests

Non-parametric vs. Parametric

Non-Parametric

- Often require much more data to train well.
- At the same time, these methods are more likely to over-fit (so we frequently combine them in an intelligent way).
- More difficult to interpret than most parametric models.
- ‘Statistical tests’ generally not important in these methods

Non-parametric vs. Parametric

- At the end of the day, it goes back to bias-variance trade off.
- Non-parametric models tend to reduce bias, while increasing variance.
- Parametric models tend to increase bias, while decreasing variance.

KNN

K-Nearest Neighbors

- Non-parametric, supervised technique

- Can be used for both regression and classification problems

- We will discuss:
  - How the algorithm works
  - What are the parameters
  - Strengths/Weaknesses

K-Nearest Neighbors

Algorithm

1. Compute the distance from each training point to your un-labeled ‘test’ data point.

2. Sort the points by distance.

3. Take the k closest points and choose the most common label (or aggregate the response value).

   (Slide 5 of Brian’s notes shows pseudo Python code)

K-Nearest Neighbors

Parameters

k… that’s it.

We also can make a decision about what distance we might use.

```
def euclidean_distance(a, b):
    return np.sqrt(np.dot(a - b, a - b))

def cosine_distance(a, b):
    return 1 - np.dot(a, b) / np.sqrt(np.dot(a, a) * np.dot(b, b))
```

K-Nearest Neighbors

Other considerations

Should we scale our variables?

What happens when we change the value of k?

How do we choose k?

What happens when we have more feature variables?

When will this method outperform other methods?

K-Nearest Neighbors

Other considerations

Should we scale our variables?
Yes

What happens when we change the value of k?
Higher k results in lower bias, higher variance.
Lower k results in higher bias, lower variance.

How do we choose k?
Cross-validation

What happens when we have more feature variables?
Curse of dimensionality - other methods will work better – nothing is close…

When will this method outperform other methods?
Outperform Linear Discriminant Analysis and Logistic Regression when the decision boundary is nonlinear. We might also use SVMs or a tree based method.

Trees