Nearest Neighbors

Prof. Eric A. Suess

February 3, 2020

Introduction

We will begin discussing **Classification** using **Nearest Neighbors**.

According to the author, nearest neighbors classifiers are defined by their classifying of unlabeled observations/examples by assigning them the class of the most similar labeled observations/examples.

k-NN algorithm

- **Training dataset** is made up of observations/examples that are classified into several categories, labeled by a nominal variable.
- **Test dataset** contains unlabeled observations/examples
- k-NN identifies k records in the training data that are the **"nearest" in similarity**.
- The unlabeled test observations/examples are assigned to the class of the majority of the k nearest neighbors.

Distance

Distance is calculated in the feature space

- **Euclidean distance**
- **Manhattan distance**

Euclidean distance

In a data set with n variables/features, the **Euclidean distance** between observations/examples is computed as follows

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

Distance Example

Distance between rows.

aa <- c(1,1)

bb <- c(2,2)

X <- rbind(aa,bb)

X

```
##    [,1] [,2]
## aa    1    1
## bb    2    2
```

Distance Example

Using the distance function in R.

dist(X)

```
##           aa
## bb 1.414214
```

Direct calculation.

sqrt(sum((aa-bb)^2))

```
## [1] 1.414214
```

Choosing k

The balance between *overfitting* and *underfitting* the *training data* is a problem known as the **bias-variance tradeoff**

Mean Squared Error

$MSE(\hat{\theta}) = Var(\hat{\theta}) + Bias^2(\hat{\theta})$

$E[(\hat{\theta} - \theta)^2] = E[(\hat{\theta} - E[\hat{\theta}])^2] + E[(E[\hat{\theta}] - \theta)^2]$

Choosing k

If k is very large, nearly every training observation/example is represented in the *final vote*, the most common training class always has a majority of voters. The model would always predict the majority class. **High Bias?**

If k is small, potentially a single nearest neighbor will determine the *final vote*, then noise may influence the prediction. **High Variance?**

The best k values is somewhere in between.

See page 71.

Preparing the data

- **min-max normalization**
  $X_{new} = \frac{X - min(X)}{max(X) - min(X)}$
- **z-score normalization**
  $X_{new} = \frac{X - \mu}{\sigma}$
- **dummy coding** for nominal variables/features

Why is the k-NN algorithm lazy?

Because no abstraction occurs. There is no model, so the method is considered to be a **non-parametric** learning method.

Example

Diagnosing *breast cancer* with k-NN algorithm.

Using R...

- loading the data
- reading the data into R
- transforming the data
- training data
- testing data
- training the model
- evaluating the model
- improving the model