Naive Bayes

Miles Erickson
(with Pinar Colak)

February 8, 2017
Galvanize, Seattle

Objectives

- **Explain why Naive Bayes is naive**
- **Describe when it is best to use Naive Bayes**
- **Explain why we need Laplace Smoothing**
- **Implement a Naive Bayes algorithm**

Motivation

Data	Labels
Email 1	spam
Email 2	not spam
.	.
.	.
Email n	not spam

Motivation

Data	Labels
Email 1	spam
Email 2	not spam
.	.
.	.
Email n	not spam

in this case

features >> observations

(p >> n)

Naive Bayes

- Computationally very efficient for high dimensions
    - Probabilities
    - No distance calculations

Goal

Classify Spam:

We want P(spam | email) or P(not spam | email)

Bayes Rule

$$P(spam|email) = \frac{P(email|spam)P(spam)}{P(email)}$$

$$P(spam) = \frac{\#\ spams}{\#\ all\ emails}$$

“Naive” Bayes

$$P(email|spam)$$

- Break down into words
- Assume independence between words (“naive”)

Naive Bayes

$$P(email|spam) = \prod_{i=1}^{p} P(word_i|spam)$$

$$P(word_i|spam) = \frac{\#\ of\ word_i\ in\ spam\ emails}{\#of\ total\ words\ in\ spam\ emails}$$

What happens if a word is not in our corpus?

Naive Bayes

Laplace smoothing:

- We want to prevent probabilities of 0
- Assume that every event happens at least α times

$$P(word_i|spam) = \frac{\#\ of\ word_i\ in\ spam\ emails + \alpha}{\#\ of\ words\ in\ spam\ emails + \alpha p}$$

- *p* is the number of features (i.e. number of words in your corpus)
- α ~ 0.01

Naive Bayes

$$P(spam|email) = P(spam) \times \prod_{i=1}^{p} P(word_i|spam)$$

$$log(P(spam|email)) =$$
$$log(P(spam)) + \sum_{i=1}^{p} log(P(word_i|spam))$$

Afternoon Assignment