



Bookmarks

- ▶ [Introduction](#)
- ▶ [Part 1: Probability and Inference](#)
- ▶ [Part 2: Inference in Graphical Models](#)
- ▼ [Part 3: Learning Probabilistic Models](#)

[Week 8: Introduction to Learning Probabilistic Models](#)

[Week 8: Introduction to Parameter Learning - Maximum Likelihood and MAP Estimation](#)

[Exercises due Nov 10, 2016 at 01:30 IST](#)



[Week 8: Homework 6](#)

[Homework due Nov 10, 2016 at 01:30 IST](#)



Part 3: Learning Probabilistic Models > Week 9: Parameter Learning - Naive Bayes Classification > The Naive Bayes Classifier: Introduction

The Naive Bayes Classifier: Introduction

[Bookmark this page](#)


THE NAIVE BAYES CLASSIFIER: INTRODUCTION (PREFACE)

Previously, we saw how maximum likelihood estimation works for some simple cases. Now, we look at how it works for a more elaborate setup, specifically in the problem of email spam detection. Note that in this section, for simplicity, in showing the maximum likelihood estimates, we will just be setting derivatives equal to 0 without checking second derivatives and boundaries.

Naive Bayes Classifier Introduction

Week 9: Parameter Learning - Naive Bayes Classification

Week 9: Mini-project on Email Spam Detection

Mini-projects due Nov 17, 2016 at 01:30 IST 

6.008.1x - Naive Bayes Classifier Intro

[Start of transcript. Skip to the end.](#)



▶ 0:00 / 0:00

▶ 1.0x



So I get a lot email spam, which makes me sad.

And so what I would like to be able to do

is to predict whether an email is spam or ham.

And "ham" is just another way of saying "not spam".

And if I could do this very accurately,

Video

[Download video file](#)

Transcripts

[Download SubRip \(.srt\) file](#)

[Download Text \(.txt\) file](#)

These notes cover roughly the same content as the video:

THE NAIVE BAYES CLASSIFIER: INTRODUCTION (COURSE NOTES)

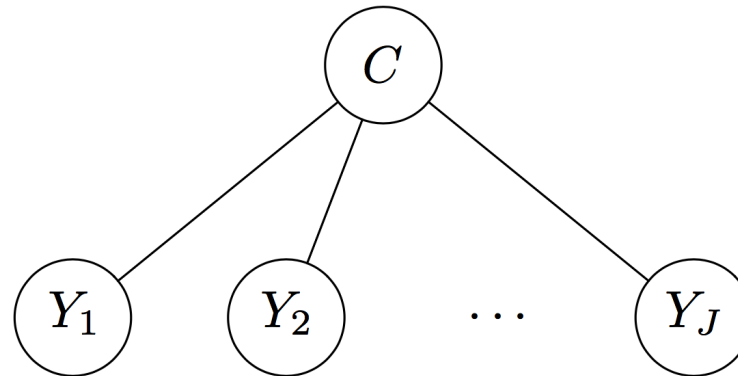
We want to build a classifier that, given an email, classifies it as either spam or ham. How do we go about doing the classification? There are many ways to classify data. Today, we're going to talk about one such way called the *naive Bayes classifier*, which uses a simple classification model and has two algorithms to go with it: the first algorithm learns the naive Bayes model parameters from training data, and the second algorithm, given the parameters learned, predicts whether a new email is spam or ham.

Suppose we have n training emails. Email i has a known label $c^{(i)} \in \{\text{spam}, \text{ham}\}$. Also, from email i , we extract J features $\mathbf{y}^{(i)} = (y_1^{(i)}, y_2^{(i)}, \dots, y_J^{(i)})$. In particular, for simplicity, we shall assume that each $y_j^{(i)} \in \{0, 1\}$ indicates the presence of the j -th word in some dictionary of J words. (Of course, you could use much fancier features such as vector-valued features or even features with non-numerical values, but we'll stick to 0's and 1's that indicate presence of certain words.) For example, maybe the first word in the dictionary is "viagra" in which case $y_1^{(i)}$ is 1 if email i contains the word "viagra" and 0 otherwise. In summary, our training data consists of $\mathbf{y}^{(1)}, \mathbf{y}^{(2)}, \dots, \mathbf{y}^{(n)} \in \{0, 1\}^J$ with respective labels $c^{(1)}, c^{(2)}, \dots, c^{(n)} \in \{\text{spam}, \text{ham}\}$.

Next, we specify a probabilistic model that explains how an email is hypothetically generated:

1. Sample a random label C that is equal to spam with probability s and equal to ham with probability $1 - s$. (For example, you could flip a coin with probability of heads s , and if the coin comes up heads, you assign $C = \text{spam}$, and otherwise you assign $C = \text{ham}$.)
2. For $j = 1, 2, \dots, J$: Sample $Y_j \sim \text{Ber}(q_j)$ if $C = \text{spam}$. Otherwise, sample $Y_j \sim \text{Ber}(p_j)$ if $C = \text{ham}$.

Note that the chance of a word from the dictionary occurring depends on whether the email is spam or ham, much like how you'd imagine that if you see the word "viagra" in an email, the email's probably spam rather than ham. The above recipe for generating features for a hypothetical email is called a *generative process*, and its corresponding graphical model is as follows:



Important observation: The Y_i 's are independent given C . This assumption of the model is not actually true for emails since certain words may be more likely to co-occur. Also, the model does not account for the ordering of the words or whether a word occurs multiple times. While the naive Bayes classifier does make these assumptions, in practice, it is often applied to data that violates these assumptions, yet the performance of the classifier is still quite good! To quote statistician George Box, "All models are wrong but some are useful."

We need to estimate the parameters $\theta = \{s, p_1, p_2, \dots, p_J, q_1, q_2, \dots, q_J\}$. We shall assume that our training data $(c^{(i)}, y^{(i)})$ for each i are generated i.i.d. according to the above generative process. Then, to learn the parameters, we find θ that maximizes the likelihood, i.e.:

$$\hat{\theta} = \arg \max_{\theta} \prod_{i=1}^n p_{C, Y_1, \dots, Y_J}(c^{(i)}, y_1^{(i)}, \dots, y_J^{(i)}; \theta).$$

Discussion

Topic: Parameter Learning - Naive Bayes Classification / The Naive Bayes Classifier: Introduction

[Show Discussion](#)

© All Rights Reserved



© 2016 edX Inc. All rights reserved except where noted. EdX, Open edX and the edX and Open EdX logos are registered trademarks or trademarks of edX Inc.

POWERED BY
OPENedX®

