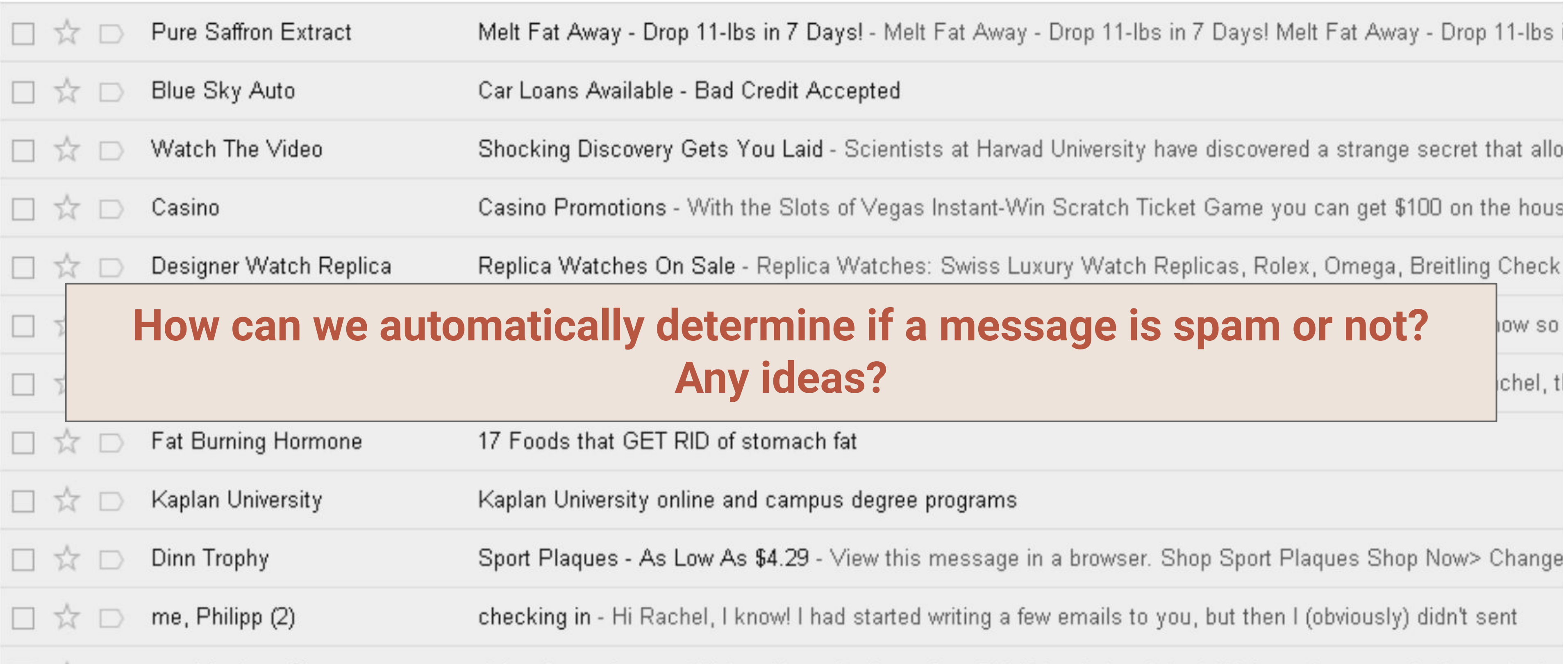
CSE 4/587
Data Intensive Computing

Dr. Eric Mikida
epmikida@buffalo.edu
208 Capen Hall

Dr. Shamshad Parvin
shamsadp@buffalo.edu
313 Davis Hall

Classifiers

Motivating Example: Spam Classification



How can we automatically determine if a message is spam or not? Any ideas?

Motivating Example: Spam Classification

Goal: Classify email into spam and not spam (binary classification)

Let's say you get an email saying "You've won the lottery!"

How do we know right away that this email is spam?

Idea: The use of certain words, ie lottery, can indicate an email is spam.

What about previous techniques?

So, our features in this problem are individual words…

Can we use linear regression or k-NN to detect spam?

What about previous techniques?

So, our features in this problem are individual words…

Can we use linear regression or k-NN to detect spam?

- Linear regression deals with continuous variables
  - We could use a heuristic to convert a continuous range into a binary range…but we are dealing with a huge number of features
- k-NN works well for low dimensionality…but again, we have a huge number of features (potentially thousands of words).
  - Curse of Dimensionality…

So what do we do?

Naive Bayes

Basic Idea: Probability of an event , based on prior knowledge of conditions that might be related to the event .

Bayes Law and Probability Theory

- Bayes' Theorem allows you to update the predicted probabilities of an event by incorporating new information.
- Bayes' Theorem was named after 18th-century mathematician Thomas Bayes.
- The theorem has become a useful element in the implementation of machine learning.

Bayes Law and Probability Theory

- For Given event x and y , we express the Bayes theorem as :

$$p(y|x) = \frac{p(x|y)p(y)}{p(x)}$$

Where ,

$p(y|x)$, probability of y given x

$p(x|y)$, probability of x given y

$p(x)$, probability of occurring event x

$p(y)$, probability of occurring event Y

Probability Theory Refresher

Here is the derivation from first principles of probabilities:

The probability of both event x and y happening , P(x,y)

- The probility of y given that x has occurred ,
- $P(y|x)=P(x,y)/P(x)$ => $P(x,y)=P(y|x)\ P(x)$ -------------(1)
- The probility of x given that y has occurred ,
- $P(x|y)=P(x,y)/P(y)$ => $P(x,y)=P\ (x|y)\ P(y)$ -------------(2)

$P(y|x)= P(x|y)\ P(y)/P(x)$ ➡

$$p(y|x) = \frac{p(x|y)p(y)}{p(x)}$$

Bayes Law – Example from Book Chapter-4

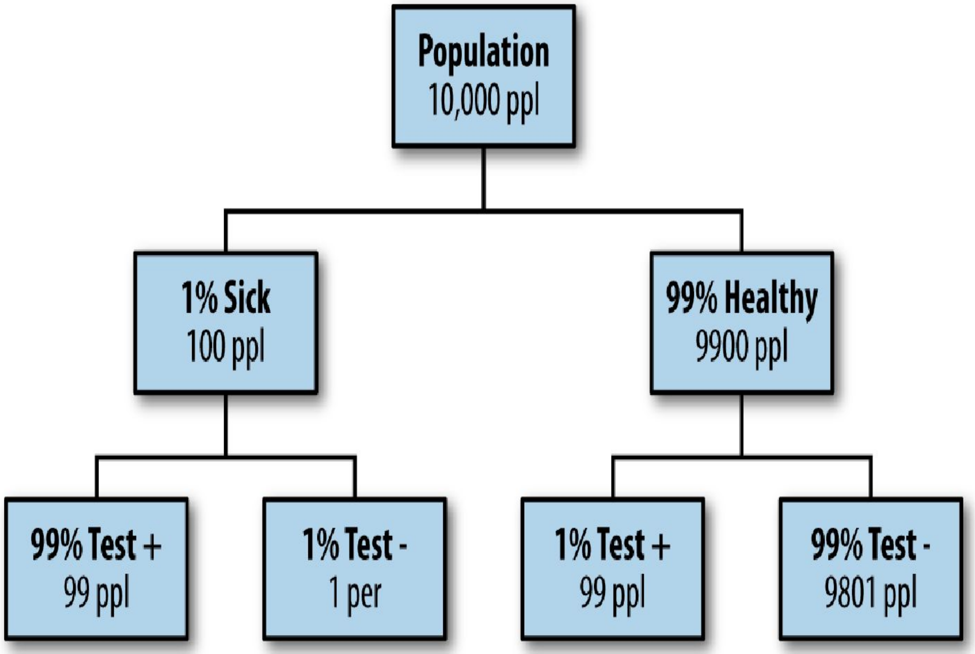
Let's say we are testing for a rare diseases where:

- 99% of sick patients test positive
- 99% of healthy patients test negative

- Given the patient test positive, what is the probability that the patient is actually sick ?

Bayes Law – Example from Book Chapter-4

- We have 100X100=10000 population
- If you test positive, you are equally likely to be healthy or sick
- Answer is 50%



Bayes Law - Example

Basic principle: $P(y \mid x) = P(x \mid y)\ P(y) / P(x)$

- **Y to refers to the event "I am sick or sick"**
- **x to refers to the event "the test is positive" or '+'**
- **Tehn we can compute**
- $P(sick|+) = \frac{P(+|sick)P(sick)}{P(+)} = \frac{0.99 \cdot 0.01}{0.99 \cdot 0.01 + 0.99 \cdot 0.01} = 0.05 = 50\%$

Bayes Law - Spam Classification

Given Bayes Law, how can we start classifying emails as spam?

Bayes Law - Spam Classification

Given Bayes Law, how can we start classifying emails as spam?

Let's start one word at a time:

$$P(spam|word) = P(word|spam) * P(spam) / P(word)$$

Bayes Law - Spam Classification

Given Bayes Law, how can we start classifying emails as spam?

Let's start one word at a time:

$$P(spam|word) = P(word|spam) * P(spam) / P(word)$$

Probability that an email is spam if it contains a given word → $P(spam|word)$

Probability that the given word appears in an email known to be spam → $P(word|spam)$

Probability that an email is spam → $P(spam)$

Probability that the given word appears in an email → $P(word)$

Bayes Law - Spam Classification

We've now boiled our classification problem down to a counting problem:

Given a set of emails that have been classified as spam or not spam (ham):

1. Count number of spam vs ham emails to compute **P(*spam*)**
2. Count number of times the given word, ie lottery, appears in emails to compute **P(*word*)**
3. Count number of times the given word appears in spam emails to compute **P(*word*|*spam*)**

Enron Email Example - DDS Chapter 4

- Enron data set containing employee emails
- 1500 spam and 3672 ham
- Test word is “meeting”
- Running a simple shell script reveals that there are 16 spam emails containing “meeting” and 153 ham emails containing "meeting"
- **Output:** What is the probability that an email containing "meeting" is spam? What is your intuition? Now prove it using Bayes Law...

Enron Email Example - DDS Chapter 4

P(*spam*) = 1500 / (1500+3672) = 0.29

Enron Email Example - DDS Chapter 4

P(*spam*) = 1500 / (1500+3672) = 0.29

P(*ham*) = 1 - **P(*spam*)** = 0.71

Enron Email Example - DDS Chapter 4

P(*spam*) = 1500 / (1500+3672) = 0.29

P(*ham*) = 1 - **P(*spam*)** = 0.71

P(*meeting|spam*) = 16/1500 = 0.0106

Enron Email Example - DDS Chapter 4

P(*spam*) = 1500 / (1500+3672) = 0.29

P(*ham*) = 1 - **P(*spam*)** = 0.71

P(*meeting|spam*) = 16/1500 = 0.0106

P(*meeting|ham*) = 153/3672 = 0.0416

Enron Email Example - DDS Chapter 4

P(*spam*) = 1500 / (1500+3672) = 0.29

P(*ham*) = 1 - **P(*spam*)** = 0.71

P(*meeting|spam*) = 16/1500 = 0.0106

P(*meeting|ham*) = 153/3672 = 0.0416

P(*meeting*) = (16+153) / (1500+3672) = 0.0326

Enron Email Example - DDS Chapter 4

$P(spam) = 1500 / (1500+3672) = 0.29$

$P(ham) = 1 - P(spam) = 0.71$

$P(meeting|spam) = 16/1500 = 0.0106$

$P(meeting|ham) = 153/3672 = 0.0416$

$P(meeting) = (16+153) / (1500+3672) = 0.0326$

$P(spam|meeting) = P(meeting|spam)*P(spam)/P(meeting) = 0.094$ (9.4%)

Enron Email Example - DDS Chapter 4

- Next we can try with other words :
- “money” : 80% chance of being spam
- “Enron”: 0% chance
- “lottery” : 1005 chance

Naive Bayes

Basic Idea: Make a probabilistic model – have many ***simple rules***, and aggregate those rules together to provide a probability.

Naive Bayes

Bayes law for each word

Basic Idea: Make a probabilistic model – have many ***simple rules***, and aggregate those rules together to provide a probability.

Naive Bayes

Bayes law for each word

Basic Idea: Make a probabilistic model – have many ***simple rules***, and aggregate those rules together to provide a probability.

Putting It All Together - Naive Bayes

So we've counted and computed probabilities for all words in our input

Putting It All Together - Naive Bayes

So we've counted and computed probabilities for all words in our input

Let's say we have ***i*** words. Let ***x*** be a vector of size ***i***,

where $\boldsymbol{x_j}$ **= 1** if the j^{th} word is present in an email, **0** otherwise.

Putting It All Together - Naive Bayes

So we've counted and computed probabilities for all words in our input

Let's say we have ***i*** words. Let ***x*** be a vector of size ***i***,

where x_j **= 1** if the j^{th} word is present in an email, **0** otherwise.

Now how do we compute P(*x*|*spam*)?

Once we do this, we can apply Bayes Law to find P(*spam*|*x*)

Naive Bayes

Let ***c*** represent the condition that an email is spam

Naive Bayes

Let ***c*** represent the condition that an email is spam

Let $x_j = 1$ if the j^{th} word is in the email

Naive Bayes

Let $\boldsymbol{c}$ represent the condition that an email is spam

Let $\boldsymbol{x_j} = \mathbf{1}$ if the $\boldsymbol{j^{th}}$ word is in the email

Let $\boldsymbol{\theta_{jc}}$ be the probability that that the $\boldsymbol{j^{th}}$ word shows up in a spam email

Naive Bayes

Let ***c*** represent the condition that an email is spam

Let $\boldsymbol{x_j}$ **= 1** if the $\boldsymbol{j^{th}}$ word is in the email

The probability that an email vector x represents a spam email looks like:

$$p(x|c) = \prod_j \theta_{jc}^{x_j}(1-\theta_{jc})^{(1-x_j)}$$

Naive Bayes

Let ***c*** represent the condition that an email is spam

Let $x_j = 1$ if the j^{th} word is in the email

Let θ_{jc} be the probability that the j^{th} word shows up in a spam email

$$p(x|c) = \prod_j \theta_{jc}^{x_j}(1 - \theta_{jc})^{(1-x_j)}$$

θ_{jc} if the j^{th} word is in the email

Naive Bayes

Let ***c*** represent the condition that an email is spam

Let $x_j = 1$ if the j^{th} word is in the email

Let θ_{jc} be the probability that the j^{th} word shows up in a spam email

$$p(x|c) = \prod_j \theta_{jc}^{x_j}(1-\theta_{jc})^{(1-x_j)}$$

θ_{jc} if the j^{th} word is in the email

$1-\theta_{jc}$ if the j^{th} word is not in the email

Example

$x = [1,1,0,0]$ $\theta_{1c} = 0.01$ $\theta_{2c} = 0.10$ $\theta_{3c} = 0.04$ $\theta_{4c} = 0.0$

Example

$\boldsymbol{x}$ = [1,1,0,0] $\theta_{1c} = 0.01$ $\theta_{2c} = 0.10$ $\theta_{3c} = 0.04$ $\theta_{4c} = 0.0$

$$p(\boldsymbol{x}|c) = \theta_{1c}\theta_{2c}(1 - \theta_{3c})(1 - \theta_{4c})$$

Example

$x = [1,1,0,0]$ $\theta_{1c} = 0.01$ $\theta_{2c} = 0.10$ $\theta_{3c} = 0.04$ $\theta_{4c} = 0.0$

$$p(x|c) = \theta_{1c}\theta_{2c}(1 - \theta_{3c})(1 - \theta_{4c})$$

$$p(x|c) = 0.01 * 0.1 * 0.96 * 1.0 = 0.00096$$

Example

$x = [1,1,0,0]$ $\theta_{1c} = 0.01$ $\theta_{2c} = 0.10$ $\theta_{3c} = 0.04$ $\theta_{4c} = 0.0$

$$p(x|\mathrm{c}) = \theta_{1c}\theta_{2c}(1 - \theta_{3c})(1 - \theta_{4c})$$

$$p(x|\mathrm{c}) = 0.01 * 0.1 * 0.96 * 1.0 = 0.00096$$

There is a 0.09% chance that this exact vector *x* appears in a spam email

Cleaning it up...

- Multiplying many small probabilities can result in numerical issues
- A common method for avoiding this is to take the log of both side

$$log(p(x|c)) = \sum_j x_j log(\theta_j/(1-\theta_j)) + \sum_j log(1-\theta_j)$$

Cleaning it up...

Many of these terms don't depend on the email and can be precomputed

$$log(p(x|c)) = \sum_j x_j log(\theta_j/(1-\theta_j)) + \sum_j log(1-\theta_j)$$

Cleaning it up...

Many of these terms don't depend on the email and can be precomputed

$$log(p(x|c)) = \sum_j x_j \boxed{log(\theta_j/(1-\theta_j))} + \sum_j log(1-\theta_j)$$

Call this $\boldsymbol{w_j}$

Cleaning it up...

Many of these terms don't depend on the email and can be precomputed

$$log(p(x|c)) = \sum_j x_j \boxed{log(\theta_j/(1-\theta_j))} + \boxed{\sum_j log(1-\theta_j)}$$

Call this w_j

Call this w_0

Cleaning it up...

Many of these terms don't depend on the email and can be precomputed

$$log(p(x|c)) = \sum_j x_j w_j + w_0$$

The Final Formula

Now given ***p*(*x*|*spam*)** we can use Baye's Law we can compute ***p*(*spam*|*x*):**

$$p(spam|x) = p(x|spam) * p(spam) / p(x)$$

The Final Formula

Now given ***p*(*x*|*spam*)** we can use Baye's Law we can compute ***p*(*spam*|*x*):**

$$p(spam|x) = p(x|spam) * p(spam) / p(x)$$

These other two terms are pretty straightforward to compute, and ***p*(*spam*)** is independent of the input email

Naive Bayes

A few notes:

- Occurrences of words are considered independent events
  - Don't care how many times a word appears
  - Don't care about combinations of words
  - This is why it's called "naive"

Naive bayes Vs K-NN

- Naive Bayes is a linear classifier, while k-NN is not.
- Curse of dimensionality and large feature sets are a problem for k-NN, while Naive Bayes performs well.

- k-NN requires no training (just load in the dataset), whereas Naive Bayes does.
- Both are examples of supervised learning (the data comes labeled).