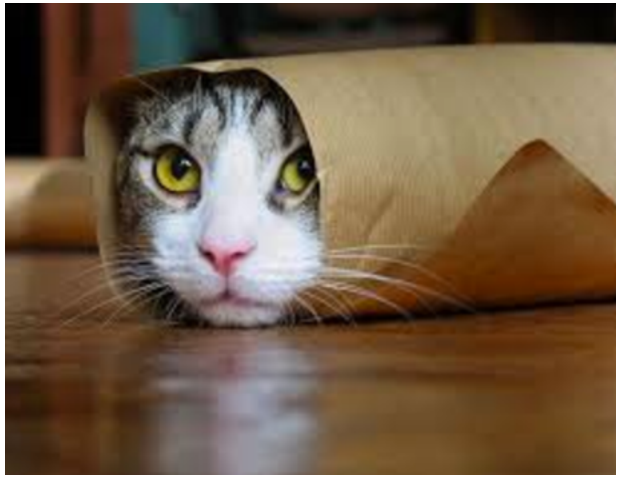
INFO-2301

Michael Paul

Feb 27, 2017

Classification

Which of these photos contains a cat?



Classification

Which of these emails are spam?

Mark Dredze
to me

Let's setup a time to talk next week.

program@emnlp2017.net via sun.s
to me

Dear Michael J. Paul,

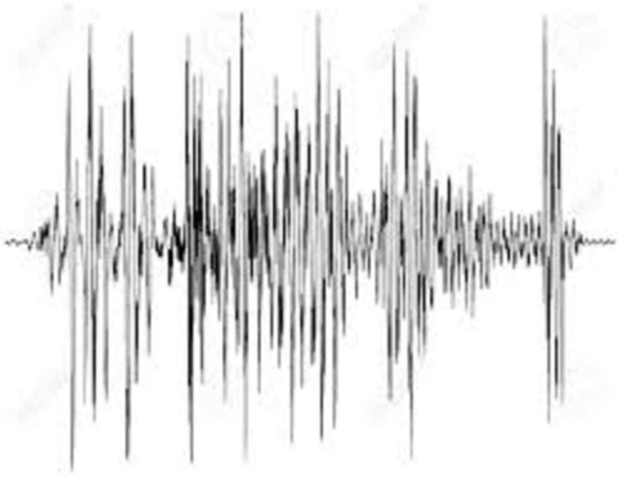
We would like to invite you to serve on the
Conference on Empirical Methods in Natu
2017), which will be held in Copenhagen,

[?]Customer-Survey[?] <u4c3pa7j8@97366ka91.frro.cvg.utn.edu.ar>
to E2M4RZEE6V

Congrats! You've Been Selected For $50 Macy's Reward

Classification

What language is this person speaking?



Classification

Assign a discrete value y to input **x**

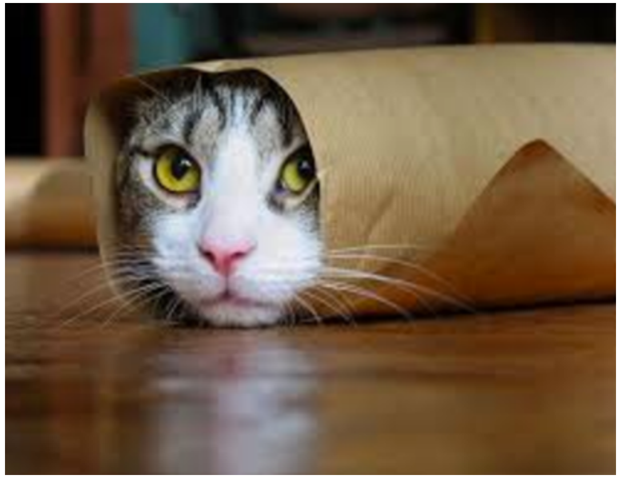
The possible values of y are called **classes**

x is usually a vector

- The dimensions of **x** correspond to **features**
- Features are properties like word counts, pixel values, etc.

Classification

Binary classification: is this photo a cat?



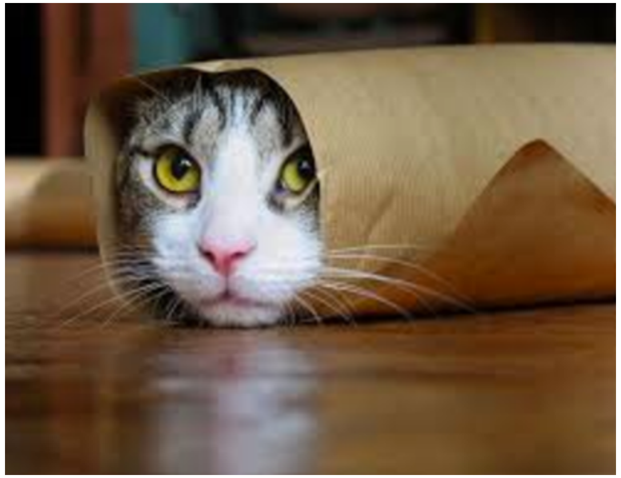
1

1

0

Classification

General classification: what kind of animal is in this photo?



cat

cat

deer

Classifiers

An algorithm that produces classifications is called a classifier

We'll learn about some common classifiers in this class

- More if you take a machine learning course

Probabilistic Classifiers

Today, we’ll look at how you can do classification with what you’ve already learned

Probabilistic Classifiers

Language modeling: recall that a 1-gram (“unigram”) language model is a discrete distribution over words

P(“you”) = 0.012

P(“the”) = 0.030

P(“said”) = 0.0015

P(“friends”) = 0.0001

Probabilistic Classifiers

Modification: *condition* the word probabilities on a class

P("you" | class="Important") = 0.0127
P("the" | class="Important") = 0.0313
P("2301" | class="Important") = 0.0021
P("winner" | class="Important") = 0.0001

P("you" | class="Spam") = 0.0201
P("you" | class="Spam") = 0.0308
P("2301" | class="Spam") = 0.0000
P("winner" | class="Spam") = 0.0150

Probabilistic Classifiers

Modification: *condition* the word probabilities on a class

P("you" | class="Important") = 0.0127
P("the" | class="Important") = 0.0313
P("2301" | class="Important") = 0.0021
P("winner" | class="Important") = 0.0001

P("you" | class="Spam") = 0.0201
P("you" | class="Spam") = 0.0308
P("2301" | class="Spam") = 0.0000
P("winner" | class="Spam") = 0.0150

Small probability in both classes. But 150 times more common in "Spam".

Probability of Text

Under a 1-gram model, what is the probability of the text sequence, "You are a winner" ?

P(w_1 = "You", w_2 = "are", w_3 = "a", w_4 = "winner")
= P(w_1 = "You") x P(w_2 = "are") x P(w_3 = "a") x P(w_4 = "winner")
= P(w = "You") x P(w = "are") x P(w = "a") x P(w = "winner")

Probability of Text

Now consider

P("You are a winner" | class="Important")

P(w_1 = "You", w_2 = "are", w_3 = "a", w_4 = "winner" | class="Important")

= P(w="You"| c="Imp.") x P(w="are"| c="Imp.") x P(w="a" | c="Imp.") x P(w="winner" | c="Imp.")

= 0.0127*0.0103*0.0285*0.0001

= 3.728085e-10

Probability of Text

Now consider

P("You are a winner" | class="Spam")

P(w_1 = "You", w_2 = "are", w_3 = "a", w_4 = "winner" | class="Spam")

= P(w="You"| c="Spam") x P(w="are"| c="Spam") x P(w="a" | c="Spam") x P(w="winner" | c="Spam")

= 0.0201*0.0141*0.0220*0.0150

= 9.35253e-8

250 times more likely to see this text when it's spam

Probability of Class

We just calculated P(text | class)

More useful for classification: P(class | text)

Bayes' rule: $$P(\text{class} \mid \text{text}) = \frac{P(\text{text} \mid \text{class})\, P(\text{class})}{P(\text{text})}$$

Probability of Class

We just calculated P(text | class)

More useful for classification: P(class | text)

Bayes' rule: $P(\text{class} \mid \text{text}) = \dfrac{P(\text{text} \mid \text{class})\, P(\text{class})}{P(\text{text})}$

This is what we calculated on the previous slide (assumes you already know the language model parameters)

Probability of Class

We just calculated P(text | class)

More useful for classification: P(class | text)

Bayes' rule: $$P(\text{class} \mid \text{text}) = \frac{P(\text{text} \mid \text{class})\ \color{red}{P(\text{class})}}{P(\text{text})}$$

This is the probability of observing a data instance from a class. For example, if 70% of your email is spam and 30% important, the P("Spam")=0.7 and P("Important")=0.3.

Probability of Class

We just calculated P(text | class)

More useful for classification: P(class | text)

Bayes' rule: $P(\text{class} \mid \text{text}) = \dfrac{P(\text{text} \mid \text{class})\, P(\text{class})}{P(\text{text})}$

You can get P(text) by *marginalization*. But as you'll see in a minute, P(text) is not important for classification because it is constant with respect to the class.

Naive Bayes

Algorithm:

1. Estimate the 1-gram language model parameters from data
    - We haven’t talked much yet about where these probabilities come from. More later.
2. For each new data instance **x**:
    1. Calculate P(class=y | **x**) for all y
    2. Return y with the largest value of P(class=y | **x**)

Naive Bayes

Bayes: Because we use Bayes’ rule

Naive: Because 1-gram models are “naïve” in that they are not a great representation of how language actually works (in the case of text)

Conditional independence:

All dimensions of **x** (e.g., all words) are independent, conditioned on the class. Because they are independent, we can use the product rule.

Naive Bayes Implementation

What you want: argmax_y P(y | x)

This is equal to: argmax_y P(x | y) P(y)

- We dropped the denominator because it doesn't depend on y. So the argmax will be the same if you just calculate the numerator.

Naive Bayes Implementation

What you want: $\text{argmax}_y\ P(y \mid x)$

This is equal to: $\text{argmax}_y\ \log(P(y \mid x))$

- This is because log is a *monotonic* function, meaning that log(x) increases as x increases, so the maximum of the log of a function will be the same as the maximum of the function.
- This will let us take advantage of an important property: $\log(a*b) = \log(a) + \log(b)$

Naive Bayes Implementation

Example: let's classify the text "You are a winner" assuming the classes are "Spam" and "Important"

Let's assume we already have all the conditional probabilities (you will be given them in your assignment).

Then we need to calculate $\log(P(x|y)P(y))$ for each y value, and return the argmax.

Naive Bayes Implementation

Score[“Important”]

= log(P(“You are a winner”|”Important”)P(“Important”))

= log(P(“You are a winner”|”Important”)) + log(P(“Important”))

= log(P(“You”|”Important”)) + log(P(“are”|”Important”)) + log(P(“a”|”Important”)) + log(P(“winner”|”Important”)) + log(P(”Important”))

Naive Bayes Implementation

Score[“Spam”]

= log(P(“You are a winner”|”Spam”)P(“Spam”))

= log(P(“You are a winner”|”Spam”)) + log(P(“Spam”))

= log(P(“You”|”Spam”)) + log(P(“are”|”Spam”)) + log(P(“a”|”Spam”)) + log(P(“winner”|” Spam”)) + log(P(” Spam”))

Naive Bayes Implementation

```
If Score["Important"] > Score["Spam"]:
        return "Important"
Else
        return "Spam"
```