CS 331: Artificial Intelligence
Naïve Bayes

Thanks to Andrew Moore for some course material

Naïve Bayes

- A special type of Bayesian network
- Makes a conditional independence assumption
- Typically used for classification

Classification

Suppose you are trying to classify situations that determine whether or not Canvas will be down. You've come up with the following list of variables (which are all Boolean):

Monday	Is a Monday
Assn	CS331 assignment due
Grades	CS331 instructor needs to enter grades
Win	The Beavers won the football game

We also have a Boolean variable called CD which stands for "Canvas down"

Classification

These are called features or attributes

This is called the "class" variable (because we're trying to classify it)

Monday	Assn	Grades	Win	CD
true	true	true	false	true
false	true	true	true	false
true	false	false	false	false
false	true	true	false	true
true	true	true	false	true
false	false	true	false	true
true	true	false	true	false

These entries in the CD column are called "class labels"

Classification

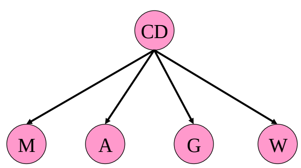
Monday	Assn	Grades	Win	CD
true	true	true	false	true
false	true	true	true	false
true	false	false	false	false
false	true	false	false	true
true	true	true	false	true
false	false	true	false	true
true	true	false	true	false

You create a dataset out of your past experience. This is called "training data".

Monday	Assn	Grades	Win
true	true	true	true
false	true	true	false

You now have 2 new situations and you would like to predict if Canvas will go down. This is called "test data".

Naïve Bayes Structure

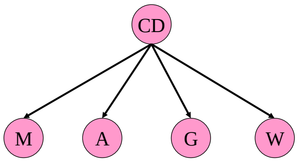


Notice the conditional independence assumption:

The features are conditionally independent given the class variable.

Naïve Bayes Parameters

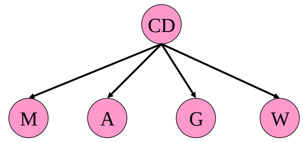
$\boldsymbol{P}(CD) = ?$



$\boldsymbol{P}(M \mid CD) = ?$ $\boldsymbol{P}(A \mid CD) = ?$ $\boldsymbol{P}(G \mid CD) = ?$ $\boldsymbol{P}(W \mid CD) = ?$

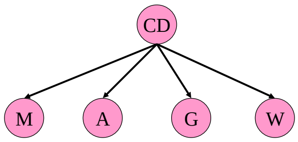
How do you get these parameters from the training data?

Naïve Bayes Parameters



CD	P(CD)
false	(# of records in training data with CD = false) / (# of records in training data)
true	(# of records in training data with CD = true) / (# of records in training data)

Naïve Bayes Parameters



M	CD	P(M | CD)
false	false	(# of records with M = false and CD = false) / (# of records with CD = false)
false	true	(# of records with M = false and CD = true) / (# of records with CD = true)
true	false	(# of records with M = true and CD = false) / (# of records with CD = false)
true	true	(# of records with M = true and CD = true) / (# of records with CD = true)

Inference in Naïve Bayes

$$\boldsymbol{P}(CD \mid M, A, G, W)$$

$$= \frac{\boldsymbol{P}(M, A, G, W \mid CD)\boldsymbol{P}(CD)}{\boldsymbol{P}(M, A, G, W)}$$

By Bayes Rule

$$= \alpha \boldsymbol{P}(M, A, G, W \mid CD)\boldsymbol{P}(CD)$$

Treat denominator as constant

$$= \alpha \boldsymbol{P}(CD)\boldsymbol{P}(M \mid CD)\boldsymbol{P}(A \mid CD)\boldsymbol{P}(G \mid CD)\boldsymbol{P}(W \mid CD)$$

From conditional independence

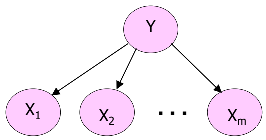
Prediction

- Suppose you are now in a day when M=true, A=true, G=true, W=true.
- You need to predict if CD=true or CD=false.
- We will use the notation that CD=true is equivalent to cd and CD=false is equivalent to ¬cd.

Prediction

- You need to compare:
  - P(cd | m, a, g, w) = α P(cd) P(m | cd) P(a | cd) P(g | cd) P(w | cd)
  - P(¬cd | m, a, g, w) = α P(¬cd) P(m | ¬cd) P(a | ¬cd) P(g | ¬cd) P(w | ¬cd)
- Whichever probability is the bigger of the two above, that is your prediction for CD
- Because you take the max of the two probabilities above, you can ignore α (since it is the same in both)

The General Case



1. Estimate P(Y=v) as fraction of records with Y=v
2. Estimate P(X_i=u | Y=v) as fraction of "Y=v" records that also have X=u.
3. To predict the Y value given observations of all the X_i values, compute

$$Y^{\text{predict}} = \underset{v}{\operatorname{argmax}}\ P(Y = v \mid X_1 = u_1 \cdots X_m = u_m)$$

13

Naïve Bayes Classifier

$$Y^{\text{predict}} = \underset{v}{\operatorname{argmax}}\ P(Y = v \mid X_1 = u_1 \cdots X_m = u_m)$$

$$Y^{\text{predict}} = \underset{v}{\operatorname{argmax}} \frac{P(Y = v, X_1 = u_1 \cdots X_m = u_m)}{P(X_1 = u_1 \cdots X_m = u_m)}$$

$$Y^{\text{predict}} = \underset{v}{\operatorname{argmax}} \frac{P(X_1 = u_1 \cdots X_m = u_m \mid Y = v) P(Y = v)}{P(X_1 = u_1 \cdots X_m = u_m)}$$

$$Y^{\text{predict}} = \underset{v}{\operatorname{argmax}}\ P(X_1 = u_1 \cdots X_m = u_m \mid Y = v) P(Y = v)$$

Because of the structure of the Bayes Net

$$Y^{\text{predict}} = \underset{v}{\operatorname{argmax}}\ P(Y = v) \prod_{j=1}^{m} P(X_j = u_j \mid Y = v)$$

14

Technical Point #1

- The probabilities P(X_j = u_j | $Y = v$) can sometimes be really small
- This can result in numerical instability since floating point numbers are not represented exactly on any computer architecture
- To get around this, use the log of the last line in the previous slide i.e.

$$Y^{\text{predict}} = \underset{v}{\operatorname{argmax}} \left[\log(P(Y = v)) + \sum_{j=1}^{m} \log(P(X_j = u_j \mid Y = v)) \right]$$

15

Technical Point #2

- When estimating parameters, what happens if you don't have any records that match a certain combination of features?
- For example, in our training data, we didn't have M=false, A=false, G=false, W=false
- This means that P(X_j = u_j | $Y = v$) in the formula below will be 0 and the entire expression will be 0.

$$P(Y = v) \prod_{j=1}^{m} P(X_j = u_j \mid Y = v)$$

Even more horrible things happen if you had this expression in log space

16

Uniform Dirichlet Priors

Let N_j be the number of values that X_j can take on.

$$P(X_j = u_j \mid V = v) = \frac{(\#\text{records with } X_j = u_j \text{ and } V = v) + 1}{(\#\text{records with } V = v) + N_j}$$

What happens when you have no records with $V = v$?

$$P(X_j = u_j \mid V = v) = \frac{1}{N_j}$$

This means that each value of X_j is equally likely in the absence of data. If you have a lot of data, it dominates the 1/N_j value. We call this trick a "uniform Dirichlet prior".

17

What You Should Know

- How to learn the parameters for a Naïve Bayes model
- How to make predictions with a Naïve Bayes model
- How to implement a Naïve Bayes Model

18

Programming Assignment #3

You will classify text into two classes.

There are two files:

1. Training data: trainingSet.txt
2. Testing data: testSet.txt

19

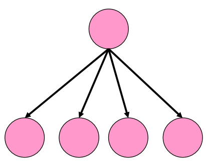
Programming Assignment #3

Two parts to this assignment:

1. Pre-processing step
2. Classification step

20

1. Preprocessing Step



- Recall that naïve Bayes has the structure shown to the right
- The nodes correspond to random variables, which are the features or attributes in the data
- What are the features in the documents?
- Note: a “document” in our assignment is a Yelp review

21

The Vocabulary

- The features of the documents will be the presence/absence of words in the vocabulary
- The vocabulary is the list of words that are known to the classifier
- Ideally, the vocabulary would be all the words in the English language
- For this assignment, you will form the vocabulary using all the words in the training data

22

Bag of Words

Suppose you have the following documents:

Training Data	Class Label
This is an excellent laptop	Class 0
No, this is not sarcasm!	Class 1

Test Data	
Excellent Laptop =P	Class 1

You will ignore punctuation for this assignment

The vocabulary will be:
this, is, an, excellent, laptop, no, not, sarcasm

23

Bag of Words

Vocab: this, is, an, excellent, laptop, no, not, sarcasm

Keep this in alphabetical order to help with debugging

Vocab: an, excellent, is, laptop, no, not, sarcasm, this

24

Training data

Next, convert your training and test data into features

Training Data

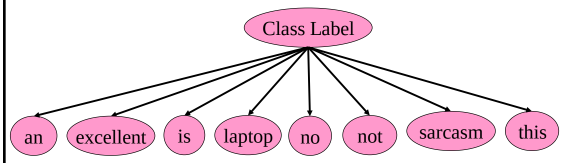
an	excellent	is	laptop	no	not	sarcasm	this	Class Label
1	1	1	1	0	0	0	1	0
0	0	1	0	1	1	1	1	1

Test Data

an	excellent	is	laptop	no	not	sarcasm	this	Class Label
0	1	0	1	0	0	0	0	1

You will output the training data in feature form, with the features alphabetized (we will grade you on this output).

2. Classification Step (Training Phase)



- Your naïve Bayes classifier now looks something like the above
- You still need to fill in the conditional probability tables in each node
- This is done in the training phase (as described on slides 9 and 10)
- Remember to use the uniform Dirichlet prior trick (see slide 17)

2. Classification Step (Testing Phase)

Testing phase

- Load the featurized test data
- For each document in the test data, predict its class label
- This requires computing:

  P(Class label | Words in document)

2. Classification Step (Testing Phase)

Suppose you have the following test instance:

an	excellent	is	laptop	no	not	sarcasm	this	Class Label
0	1	0	1	0	0	0	0	(to be predicted)

$$
\begin{aligned}
&P(Class = 1 \mid an = 0, excellent = 1, is = 0, laptop = 1, no = 0, not \\
&= 0, sarcasm = 0, this = 0) \\
&= \alpha P(Class = 1) * P(an = 0|Class = 1) * P(excellent = 1|Class = 1) * \\
&\quad P(is = 0|Class = 1) * P(laptop = 1|Class = 1) * P(no=0|Class =1) * \\
&\quad P(not = 0|Class = 1) * P(sarcasm = 0|Class = 1) * \\
&\quad P(this = 0|Class = 1)
\end{aligned}
$$

Note: Use P(Word = 1 | Class) if you have a 1 for the word. Otherwise use P(Word = 0 | Class)

2. Classification Step (Testing Phase)

an	excellent	is	laptop	no	not	sarcasm	this	Class Label
0	1	0	1	0	0	0	0	(to be predicted)

Then compute the following:

$$
\begin{aligned}
&P(Class = 0 \mid an = 0, excellent = 1, is = 0, laptop = 1, no = 0, not \\
&= 0, sarcasm = 0, this = 0) \\
&= \alpha P(Class = 0) * P(an = 0|Class = 0) * P(excellent = 1|Class = 0) * \\
&\quad P(is = 0|Class = 0) * P(laptop = 1|Class = 0) * P(no=0|Class =0) * \\
&\quad P(not = 0|Class = 0) * P(sarcasm = 0|Class = 0) * \\
&\quad P(this = 0|Class = 0)
\end{aligned}
$$

2. Classification Step (Testing Phase)

an	excellent	is	laptop	no	not	sarcasm	this	Class Label
0	1	0	1	0	0	0	0	(to be predicted)

If

$$
\begin{aligned}
&P(Class = 1 \mid an = 0, excellent = 1, is = 0, laptop \\
&= 1, no = 0, not = 0, sarcasm = 0, this = 0) \\
&> \\
&P(Class = 0 \mid an = 0, excellent = 1, is = 0, laptop \\
&= 1, no = 0, not = 0, sarcasm = 0, this = 0)
\end{aligned}
$$

Predict Class = 1 otherwise predict Class = 0

2. Classification Step (Testing Phase)

- For each document in the testing data set, predict its class label
- Compare the predicted class label to the actual class label
- Output the accuracy for each class:

$$\frac{\text{\# correctly predicted class labels}}{\text{total \# of predictions}}$$

Results

There are two sets of results we require:

1. Results #1:
   - Use trainingSet.txt for the training phase
   - Use trainingSet.txt for the testing phase
   - Report accuracy
2. Results #2:
   - Use trainingSet.txt for the training phase
   - Use testSet.txt for the testing phase
   - Report accuracy