NAÏVE BAYES

David Kauchak
CS159 Spring 2019

Admin

Assignment 7 out soon (due next Friday at 5pm)

Quiz #3 next Monday
- Text similarity -> this week (though, light on ML)

Final project

Final project

1. Your project should relate to something involving NLP

2. Your project must include a solid experimental evaluation

3. Your project should be in a pair or group of three. If you'd like to do it solo or in a group of four, please come talk to me.

Final project

<table>
<thead>
<tr>
<th>date</th>
<th>time</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>4/17</td>
<td>in-class</td>
<td>Project proposal presentation</td>
</tr>
<tr>
<td>4/21</td>
<td>11:50pm</td>
<td>Project proposal write-up</td>
</tr>
<tr>
<td>4/28</td>
<td>11:50pm</td>
<td>Status report</td>
</tr>
<tr>
<td>5/3</td>
<td>5pm</td>
<td>Paper draft</td>
</tr>
<tr>
<td>5/8</td>
<td>in-class</td>
<td>Final paper, code and presentation</td>
</tr>
</tbody>
</table>

Read the final project handout ASAP!

Start forming groups and thinking about what you want to do
Final project ideas

- pick a text classification task
  - evaluate different machine learning methods
  - implement a machine learning method
  - analyze different feature categories

- n-gram language modeling
  - implement and compare different smoothing techniques
  - implement alternative models

- parsing
  - lexicalized PCFG (with smoothing)
  - n-best list generation
  - parse output reranking
  - implement another parsing approach and compare
  - parsing non-traditional domains (e.g. Twitter)

- EM
  - try and implement EM model 2
  - word-level translation models

Final project application areas

- spelling correction
- part of speech tagger
- text chunker
- dialogue generation
- pronoun resolution
- compare word similarity measures (more than the ones we looked at)
- word sense disambiguation
- machine translation
- information retrieval
- information extraction
- question answering
- summarization
- speech recognition

Probabilistic Modeling

1. Model the data with a probabilistic model
   - specifically, learn \( p(\text{features, label}) \)
   - \( p(\text{features, label}) \) tells us how likely these features and this example are

Basic steps for probabilistic modeling

1. Pick a model
2. Figure out how to estimate the probabilities for the model
3. (Optional): deal with overfitting

Probabilistic models

Which model do we use, i.e., how do we calculate \( p(\text{feature, label}) \)?

How do we train the model, i.e., how do we estimate the probabilities for the model?

How do we deal with overfitting?
Naïve Bayes assumption

\[ p(\text{features, label}) = p(y) \prod_{j=1}^{m} p(x_j | y, x_1, ..., x_{j-1}) \]

\[ p(x_j | y, x_1, ..., x_{j-1}) = p(x_j | y) \]

What does this assume?

Naïve Bayes model

\[ p(\text{features, label}) = p(y) \prod_{j=1}^{m} p(x_j | y, x_1, ..., x_{j-1}) \]

\[ p(x_j | y, x_1, ..., x_{j-1}) = p(x_j | y) \]

Assumes feature \( j \) is independent of the other features given the label.

How do we model this?
- for binary features (e.g., “banana” occurs in the text)
- for discrete features (e.g., “banana” occurs \( x \) times)
- for real valued features (e.g., the text contains \( x \) proportion of verbs)

\[ p(x | y) \]

Binary features (aka, Bernoulli Naïve Bayes):

\[ p(x_j | y) = \begin{cases} \theta_j & \text{if } x_j = 1 \\ 1 - \theta_j & \text{otherwise} \end{cases} \]

biased coin toss!
Basic steps for probabilistic modeling

Step 1: pick a model

Step 2: figure out how to estimate the probabilities for the model

Step 3 (optional): deal with overfitting

**Probabilistic models**

Which model do we use, i.e. how do we calculate $p(\text{feature, label})$?

How do we train the model, i.e. how do we estimate the probabilities for the model?

How do we deal with overfitting?

**Obtaining probabilities**

$$p(\text{y}) \prod_{j=1}^{m} p(x_j | y)$$

[ $m$ = number of features]

**MLE estimation for Bernoulli NB**

$$p(y) \prod_{j=1}^{m} p(x_j | y)$$

$$p(y) = \frac{\text{count}(y)}{n}$$  
number of examples with label  
number of total examples

$$p(x_j | y) = \frac{\text{count}(x_j, y)}{\text{count}(y)}$$  
number of examples with the label with feature  
number of examples with label

What are the MLE estimates for these?

What does training a NB model then involve?  
How difficult is this to calculate?
Text classification

\[ p(y) = \frac{\text{count}(y)}{n} \]

\[ p(w_j \mid y) = \frac{\text{count}(w_j, y)}{\text{count}(y)} \]

Unigram features: \(w_j\), whether or not word \(w_j\) occurs in the text

What are these counts for text classification with unigram features?

Naive Bayes classification

\[ p(y) = \frac{\text{count}(y)}{n} \]

\[ p(w_j \mid y) = \frac{\text{count}(w_j, y)}{\text{count}(y)} \]

<table>
<thead>
<tr>
<th>Feature</th>
<th>Count of (y)</th>
<th>Total number of texts</th>
<th>Count of (y) with (w_j)</th>
<th>Count of (y) without (w_j)</th>
</tr>
</thead>
<tbody>
<tr>
<td>yellow, curved, no leaf, 6oz, banana</td>
<td>0.004</td>
<td>number of texts with label</td>
<td>number of texts with the label with word (w_j)</td>
<td>number of texts with label</td>
</tr>
</tbody>
</table>

Given an unlabeled example: yellow, curved, no leaf, 6oz predict the label

How do we use a probabilistic model for classification/prediction?

NB classification

\[ p(y) = \frac{\text{count}(y)}{n} \]

\[ p(w_j \mid y) = \frac{\text{count}(w_j, y)}{\text{count}(y)} \]

**Probabilistic model:**

\[ p_y = \prod_{j=1}^{n} p(x_j \mid y) \]

\[ p_{\text{argmax}} = \prod_{j=1}^{n} p(x_j \mid y) \]

Predict largest

label = \[ \text{argmax}_{y \in \text{labels}} p(y) \prod_{j=1}^{n} p(x_j \mid y) \]
NB classification

**Probabilistic Model:**

\[ p(\text{features, label}) \]

- **Yellow, curved, no leaf, 6oz, banana**
- **Yellow, curved, no leaf, 6oz, apple**

Notice that each label has its own separate set of parameters, i.e. \( p(x_j | y) \)

Bernoulli NB for text classification

**Probabilistic Model:**

\[ p(\text{features, label}) \]

\[ p(y = 1) \]

\[ p(x_j | y = 1) \]

\[ \prod_{j | y = 1} \]

pick largest

\[ p(y = 2) \]

\[ p(x_j | y = 2) \]

\[ \prod_{j | y = 2} \]

pick largest

How good is this model for text classification?

For text classification, what is this computation? Does it make sense?

Bernoulli NB for text classification

\[ (1, 1, 0, 0, 0, 1, 0, 0, ...) \]

\[ p(y = 1) \]

\[ p(w_j | y = 1) \]

\[ \prod_{j | y = 1} \]

pick largest

\[ p(y = 2) \]

\[ p(w_j | y = 2) \]

\[ \prod_{j | y = 2} \]

pick largest

Each word that occurs, contributes \( p(w_j | y) \)

Each word that does NOT occur, contributes 1 - \( p(w_j | y) \)
Generative Story

To classify with a model, we’re given an example and we obtain the probability

We can also ask how a given model would generate an example

This is the “generative story” for a model

Looking at the generative story can help understand the model

We also can use generative stories to help develop a model

Bernoulli NB generative story

$p(y) \prod_{j=1}^{k} p(x_j | y)$

What is the generative story for the NB model?

Bernoulli NB generative story

1. Pick a label according to $p(y)$
   - roll a biased, num_labels-sided die

2. For each feature:
   - Flip a biased coin:
     - if heads, include the feature
     - if tails, don’t include the feature

   What does this mean for text classification, assuming unigram features?

Bernoulli NB generative story

1. Pick a label according to $p(y)$
   - roll a biased, num_labels-sided die

2. For each word in your vocabulary:
   - Flip a biased coin:
     - if heads, include the word in the text
     - if tails, don’t include the word
Bernoulli NB

\[ p(y) \prod_{j=1}^{m} p(x_j | y) \]

Pros/cons?

Pros
- Easy to implement
- Fast!
- Can be done on large data sets

Cons
- Naïve Bayes assumption is generally not true
- Performance isn’t as good as other models
- For text classification (and other sparse feature domains) the \( p(x_i = 0 | y) \) can be problematic

Another generative story

Randomly draw words from a “bag of words” until document length is reached

Draw words from a fixed distribution

Selected: \( w_2 \)
Draw words from a fixed distribution

Selected: \(w_1\)

Put a copy of \(w_1\) back

Draw words from a fixed distribution

Selected: \(w_1, w_3\)

Draw words from a fixed distribution

Selected: \(w_1, w_1\)

Put a copy of \(w_1\) back
Draw words from a fixed distribution

Selected: $w_1 \ w_3 \ w_2$

Put a copy of $w_2$ back

Draw words from a fixed distribution

Selected: $w_1 \ w_2 \ w_2 \ ...$

Draw words from a fixed distribution

Is this a NB model, i.e. does it assume each individual word occurrence is independent?

Draw words from a fixed distribution

Yes! Doesn’t matter what words were drawn previously, still the same probability of getting any particular word
Draw words from a fixed distribution

Does this model handle multiple word occurrences?

Selected: $w_1, w_2, w_3, \ldots$

NB generative story

1. Pick a label according to $p(y)$
   - roll a biased, $num_labels$-sided die
2. For each word in your vocabulary:
   - Flip a biased coin:
     - If heads, include the word in the text
     - If tails, don’t include the word

Bernoulli NB

Multinomial NB

Probabilities

1. Pick a label according to $p(y)$
   - roll a biased, $num_labels$-sided die
2. For each word in your vocabulary:
   - Flip a biased coin:
     - If heads, include the word in the text
     - If tails, don’t include the word
3. Keep drawing words from $p(\text{words} | y)$ until text length has been reached

$P(y) \prod_{j=1}^{m} p(x_j | y)$

$\{1, 1, 1, 0, 0, 1, 0, 0, \ldots\}$

$\{4, 1, 2, 0, 0, 7, 0, 0, \ldots\}$

$w_1, w_2, w_3, \ldots$
A digression: rolling dice

What's the probability of getting a 3 for a single roll of this dice?

1/6

A digression: rolling dice

What is the probability distribution over possible single rolls?

1/6 1/6 1/6 1/6 1/6 1/6
1 2 3 4 5 6

A digression: rolling dice

What if I told you 1 was twice as likely as the others?

2/7 1/7 1/7 1/7 1/7 1/7
1 2 3 4 5 6

A digression: rolling dice

What if I rolled 400 times and got the following number?

1: 100
2: 50
3: 50
4: 100
5: 50
6: 50

1/4 1/8 1/8 1/4 1/8 1/8
1 2 3 4 5 6
A digression: rolling dice

1. What is the probability of rolling a 1 and a 5 (in any order)?
2. Two 1s and a 5 (in any order)?
3. Five 1s and two 5s (in any order)?

\[
\frac{1}{4} \quad \frac{1}{8} \quad \frac{1}{8} \quad \frac{1}{8} \quad \frac{1}{8} \quad \frac{1}{8}
\]

Multinomial distribution

Multinomial distribution: independent draws over \(m\) possible categories

If we have frequency counts \(x_1, x_2, \ldots, x_m\) over each of the categories, the probability is:

\[
p(x_1, x_2, \ldots, x_m | \theta_1, \theta_2, \ldots, \theta_m) = \frac{n!}{\prod_{j=1}^{m} x_j!} \prod_{j=1}^{m} \theta_j^{x_j}
\]

where

- \(n\) is the total number of observations
- \(x_j\) is the count of the \(j\)th category
- \(\theta_j\) is the probability of the \(j\)th category

What are \(\theta_j\)?
Are there any constraints on the values that they can take?
Multinomial distribution

\[ p(x_1, x_2, \ldots, x_n | \theta_1, \theta_2, \ldots, \theta_m) = \frac{n!}{\prod_{j=1}^{n} x_j!} \prod_{j=1}^{m} \theta_j^{x_j} \]

- \( \theta_j \): probability of rolling "j"
- \( \theta_j \geq 0 \)
- \( \sum_{j=1}^{m} \theta_j = 1 \)

\( \theta_1, \theta_2, \ldots, \theta_m \)

1 2 3 4 5 6 ...  

Why the digression?

\[ p(x_1, x_2, \ldots, x_n | \theta_1, \theta_2, \ldots, \theta_m) = \frac{n!}{\prod_{j=1}^{n} x_j!} \prod_{j=1}^{m} \theta_j^{x_j} \]

Drawing words from a bag is the same as rolling a die!

number of sides = number of words in the vocabulary

Back to words...

Basic steps for probabilistic modeling

Model each class as a multinomial:

\[ p(\text{features}, \text{label}) = p(y) \frac{n!}{\prod_{j=1}^{n} \prod_{i=1}^{m} \theta_i^{x_{ij}}} \]

Step 2: figure out how to estimate the probabilities for the model

How do we train the model, i.e. estimate \( \theta_j \) for each class?
A digression: rolling dice

What if I rolled 400 times and got the following number?

1: 100
2: 50
3: 50
4: 100
5: 50
6: 50

1/4 1/8 1/8 1/4 1/8 1/8
1 2 3 4 5 6

Training a multinomial

For each label, y:

\( w_1: 100 \) times
\( w_2: 50 \) times
\( w_3: 10 \) times
\( w_4: \ldots \)

\[ \theta_j = \frac{\text{count}(w_j, y)}{\sum \text{count}(w_i, y)} \]

\( = \frac{\text{number of times word } w_j \text{ occurs in label } y \text{ docs}}{\text{total number of words in label } y \text{ docs}} \)

1/4 1/8 1/8 1/4 1/8 1/8
1 2 3 4 5 6

Classifying with a multinomial

\( p(y=1) = \frac{\prod_{j=1}^{m} \theta_j^{x_j}}{\prod_{j=1}^{m} \theta_j^{x_j}} \)

\( p(y=2) = \frac{\prod_{j=1}^{m} \theta_j^{x_j}}{\prod_{j=1}^{m} \theta_j^{x_j}} \)

Any way I can make this simpler?

pick largest
Classifying with a multinomial

\( \{10, 2, 6, 0, 1, 0, 0, \ldots\} \)

\( p(y=1) = \prod_{j=1}^{n} \theta_j^{x_{ij}} \)

\( p(y=2) = \prod_{j=1}^{n} \theta_j^{x_{ij}} \)

\( \frac{n!}{\prod_{j=0}^{m_j}} \) is a constant!

pick largest

Multinomial finalized

Training:
- Calculate \( p(\text{label}) \)
- For each label, calculate \( \theta \)

\( \theta_j = \frac{\text{count}(w_j, y)}{\sum_{y=1}^{m} \text{count}(w_j, y)} \)

Classification:
- Get word counts
- For each label you had in training, calculate:

\[ p(y) \prod_{j=1}^{m} \theta_j^{x_{ij}} \]

and pick the largest

Multinomial vs. Bernoulli?

Handles word frequency

Given enough data, tends to performs better

Multinomial vs. Bernoulli?

Handles word frequency

Given enough data, tends to performs better
Multinomial vs. Bernoulli?

- Handles word frequency
- Given enough data, tends to perform better

Maximum likelihood estimation

- Intuitive
- Sets the probabilities so as to maximize the probability of the training data

Problems?
- Overfitting!
- Amount of data
- Particularly problematic for rare events
- Is our training data representative

Basic steps for probabilistic modeling

1. Pick a model
2. Figure out how to estimate the probabilities for the model
3. (Optional) Deal with overfitting

Unseen events

Probabilistic models

- Which model do we use, i.e. how do we calculate p(feature, label)?
- How do we train the model, i.e. how do we estimate the probabilities for the model?
- How do we deal with overfitting?

Unseen event example:

- Positive: banana: 2
- Negative: banana: 0

θ_j = \frac{\text{count}(w_j, y)}{\sum \text{count}(w_j, y)}

What will θ_{banana} be for the negative class?
Unseen events

\[ \theta_j = \frac{\text{count}(w_j, y)}{\sum_{i=1}^{n} \text{count}(w_i, y)} \]

What will \( \theta_{\text{banana}} \) be for the negative class?

\( 0! \) Is this a problem?

Add lambda smoothing

\[ \theta_j = \frac{\text{count}(w_j, y) + \lambda}{\lambda n + \sum_{i=1}^{n} \text{count}(w_i, y)} \]

for each label, pretend like we've seen each feature/word occur in \( \lambda \) additional examples
Different than…

How is this problem different?

Out of vocabulary. Many ways to solve… for our implementation, we’ll just ignore them.