Probabilistic Modeling

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

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

How do we deal with overfitting?
Naïve Bayes assumption

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

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

What does this assume?

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

Naïve Bayes model

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

\[ p(x_j | y) = \frac{\theta_j}{1 - \theta_j} \]

\( \theta_j \) is the probability of a particular feature value 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)

Other features types:
- Could use a lookup table for each value, but doesn't generalize well
- Better, model as a distribution:
  - gaussian (i.e. normal) distribution
  - poisson distribution
  - multinomial distribution (more on this later)
  - …
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 \( p(\text{feature}, \text{label}) \)?

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

How do we deal with overfitting?

Obtaining probabilities

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

Training data

MLE estimation for NB

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

\[
p(x_j | \mathbf{y}) = \frac{\text{count}(x_j, \mathbf{y})}{\text{count}(\mathbf{y})}
\]

What are the MLE estimates for these?

Maximum likelihood estimates

What does training a NB model then involve?

How difficult is this to calculate?
Text classification

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

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

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

What are these counts for text classification with unigram features?

Naïve Bayes classification

\[ p(y) = \prod_{j \in \text{features}} p(x_j | y) \]

Given an unlabeled example: yellow, curved, no leaf, 6oz, banana
- Predict the label

Naïve Bayes classification

\[ p(y) = \prod_{j \in \text{features}} p(x_j | y) \]

Given an unlabeled example: yellow, curved, no leaf, 6oz, banana
- Predict the label

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

**NB classification**

\[ p(y) = \prod_{j \in \text{features}} p(x_j | y) \]

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

\[ p(y = 2) \prod_{j} p(x_j | y = 2) \]

pick largest

\[ \text{label} = \arg \max_{\text{labels}} p(y) \prod_{j} p(x_j | y) \]
NB classification

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

Bernoulli NB for text classification

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

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

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

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 more complicated 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: \( w_1 \)
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: \(w_3, w_2\)

Put a copy of \(w_1\) back

Draw words from a fixed distribution

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

Draw words from a fixed distribution

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

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 \ ...

NB generative story

<table>
<thead>
<tr>
<th>Bernoulli NB</th>
<th>Multinomial NB</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Pick a label according to $p(y)$: roll a biased, num_labels-sided die</td>
<td>1. Pick a label according to $p(y)$: roll a biased, num_labels-sided die</td>
</tr>
<tr>
<td>2. For each word in your vocabulary:</td>
<td>2. For each word in your vocabulary:</td>
</tr>
<tr>
<td>- Flip a biased coin:</td>
<td>- Flip a biased coin:</td>
</tr>
<tr>
<td>- If heads, include the word in the text</td>
<td>- If heads, include the word in the text</td>
</tr>
<tr>
<td>- If tails, don’t include the word</td>
<td>- If tails, don’t include the word</td>
</tr>
<tr>
<td>3. Keep drawing words from $p(\text{words}</td>
<td>y)$ until text length has been reached.</td>
</tr>
</tbody>
</table>

Probabilities

<table>
<thead>
<tr>
<th>Bernoulli NB</th>
<th>Multinomial NB</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Pick a label according to $p(y)$: roll a biased, num_labels-sided die</td>
<td>1. Pick a label according to $p(y)$: roll a biased, num_labels-sided die</td>
</tr>
<tr>
<td>2. For each word in your vocabulary:</td>
<td>2. Keep drawing words from $p(\text{words}</td>
</tr>
<tr>
<td>- Flip a biased coin:</td>
<td>$p(y) \prod_{j=1}^{m} p(x_j</td>
</tr>
<tr>
<td>- If heads, include the word in the text</td>
<td>$(3, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0)$</td>
</tr>
<tr>
<td>- If tails, don’t include the word</td>
<td>$(4, 1, 2, 0, 0, 7, 0, 0, 0, 0, 0)$</td>
</tr>
<tr>
<td>$p(x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11})$</td>
<td>$p(x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11})$</td>
</tr>
</tbody>
</table>
A digression: rolling dice

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

\[ \frac{1}{6} \]

A digression: rolling dice

What is the probability distribution over possible single rolls?

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1/6</td>
<td>1/6</td>
<td>1/6</td>
<td>1/6</td>
<td>1/6</td>
<td>1/6</td>
</tr>
</tbody>
</table>

A digression: rolling dice

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

\[ \frac{2}{7} \quad \frac{1}{7} \quad \frac{1}{7} \quad \frac{1}{7} \quad \frac{1}{7} \quad \frac{1}{7} \]

A digression: rolling dice

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

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>100</td>
<td>50</td>
<td>50</td>
<td>100</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>1/4</td>
<td>1/8</td>
<td>1/8</td>
<td>1/4</td>
<td>1/8</td>
<td>1/8</td>
<td>1/8</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td></td>
</tr>
</tbody>
</table>
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)?

\[
\begin{array}{cccccc}
1/4 & 1/8 & 1/8 & 1/4 & 1/8 & 1/8 \\
1 & 2 & 3 & 4 & 5 & 6 \\
\end{array}
\]

Multinomial distribution

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

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

number of different ways to get those counts

probability of particular counts

\(\theta_j\) are the parameters, are there any constraints on the values that they can take?

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

General formula?

\[
p(x_1, x_2, \ldots, x_m | \theta_1, \theta_2, \ldots, \theta_m) = \frac{n!}{\prod x_j!} \prod \theta_j^{x_j}
\]
Multinomial distribution

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

\( \theta_j \): probability of rolling "j"

\[ \theta_j \geq 0 \]

\[ \sum_{j=1}^{m} \theta_j = 1 \]

\( \theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6, \ldots \)

Back to words...

Why the digression?

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

number of sides = number of words in the vocabulary

Basic steps for probabilistic modeling

Model each class as a multinomial:

\[ p(features, label) = p(y) \frac{n!}{s_y! \cdot \sum_{j=1}^{s_y} \theta_y^{j}} \]

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

How do we train the model, i.e. estimate \( \theta \) 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

Training a multinomial

For each label, $y$:

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

where $w$ is the word, $y$ is the label, and $\text{count}(w, y)$ is the number of times word $w$ occurs in label $y$ docs.

Classifying with a multinomial

\[
p(y=1) = \frac{n!}{\prod \theta_j^{x_j}}
\]

\[
p(y=2) = \frac{n!}{\prod \theta_j^{x_j}}
\]

Any way I can make this simpler?
Classifying with a multinomial

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

\[
p(y = 2) = \prod_{j=1}^{m} \theta_j \]

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

\[
p(y = 1) \quad \text{pick largest} \]

\[
p(y = 2) \quad \text{pick largest} \]

\[
p(y = \ldots) \quad \text{pick largest} \]

Multinomial finalized

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

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

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

\[
p(y = 1) \quad \text{pick largest}
\]

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 performs better