Classification: Naive Bayes and Logistic Regression

Natural Language Processing: Jordan Boyd-Graber
University of Colorado Boulder
SEPTEMBER 17, 2014

Slides adapted from Hinrich Schütze and Lauren Hannah
By the end of today …

- You’ll be able to frame many standard NLP tasks as classification problems
- Apply logistic regression (given weights) to classify data
- Learn naïve Bayes from data
Outline

1 Classification
2 Logistic Regression
3 Logistic Regression Example
4 Motivating Naïve Bayes Example
5 Naive Bayes Definition
6 Wrapup
Formal definition of Classification

Given:

- A universe \( X \) our examples can come from (e.g., English documents with a predefined vocabulary)
Formal definition of Classification

Given:

- A universe $X$ our examples can come from (e.g., English documents with a predefined vocabulary)
  - Examples are represented in this space. (e.g., each document has some subset of the vocabulary; more in a second)
Formal definition of Classification

Given:

- A universe $\mathbb{X}$ our examples can come from (e.g., English documents with a predefined vocabulary)
  - Examples are represented in this space. (e.g., each document has some subset of the vocabulary; more in a second)
- A fixed set of classes $\mathbb{C} = \{c_1, c_2, \ldots, c_J\}$
**Formal definition of Classification**

Given:

- A universe $\mathbb{X}$ our examples can come from (e.g., English documents with a predefined vocabulary)
  - Examples are represented in this space. (e.g., each document has some subset of the vocabulary; more in a second)
- A fixed set of classes $\mathbb{C} = \{c_1, c_2, \ldots, c_J\}$
  - The classes are human-defined for the needs of an application (e.g., spam vs. ham).
Formal definition of Classification

Given:

- A universe $X$ our examples can come from (e.g., English documents with a predefined vocabulary)
  - Examples are represented in this space. (e.g., each document has some subset of the vocabulary; more in a second)
- A fixed set of classes $C = \{c_1, c_2, \ldots, c_J\}$
  - The classes are human-defined for the needs of an application (e.g., spam vs. ham).
- A training set $D$ of labeled documents with each labeled document $d \in X \times C$
Formal definition of Classification

Given:

- A universe $X$ our examples can come from (e.g., English documents with a predefined vocabulary)
  - Examples are represented in this space. (e.g., each document has some subset of the vocabulary; more in a second)
- A fixed set of classes $C = \{c_1, c_2, \ldots, c_J\}$
  - The classes are human-defined for the needs of an application (e.g., spam vs. ham).
- A training set $D$ of labeled documents with each labeled document $d \in X \times C$

Using a learning method or learning algorithm, we then wish to learn a classifier $\gamma$ that maps documents to classes:

$$\gamma : X \rightarrow C$$
Classification

Topic classification

classes:
- UK
- China
- poultry
- coffee
- elections
- sports

training set:
- "congestion"
- "London"
- "Olympics"
- "Beijing"
- "feed"
- "chicken"
- "roasting"
- "beans"
- "recount"
- "votes"
- "diamond"
- "baseball"
- "Parliament"
- "Big Ben"
- "tourism"
- "Great Wall"
- "pate"
- "ducks"
- "arabica"
- "robusta"
- "seat"
- "run-off"
- "forward"
- "soccer"
- "Windsor"
- "the Queen"
- "Mao"
- "communist"
- "bird flu"
- "turkey"
- "Kenya"
- "harvest"
- "TV ads"
- "campaign"
- "team"
- "captain"

γ(d') = China

d'

test set:
- "first"
- "private"
- "Chinese"
- "airline"
Examples of how search engines use classification

- Standing queries (e.g., Google Alerts)
- Language identification (classes: English vs. French etc.)
- The automatic detection of spam pages (spam vs. nonspam)
- The automatic detection of sexually explicit content (sexually explicit vs. not)
- Sentiment detection: is a movie or product review positive or negative (positive vs. negative)
- Topic-specific or vertical search – restrict search to a “vertical” like “related to health” (relevant to vertical vs. not)
Classification methods: 1. Manual

• Manual classification was used by Yahoo in the beginning of the web. Also: ODP, PubMed
• Very accurate if job is done by experts
• Consistent when the problem size and team is small
• Scaling manual classification is difficult and expensive.
• → We need automatic methods for classification.
Classification methods: 2. Rule-based

- There are “IDE” type development environments for writing very complex rules efficiently. (e.g., Verity)
- Often: Boolean combinations (as in Google Alerts)
- Accuracy is very high if a rule has been carefully refined over time by a subject expert.
- Building and maintaining rule-based classification systems is expensive.
Classification methods: 3. Statistical/Probabilistic

- As per our definition of the classification problem – text classification as a learning problem
- Supervised learning of a the classification function $\gamma$ and its application to classifying new documents
- We will look at a couple of methods for doing this: Naive Bayes, Logistic Regression, SVM, Decision Trees
- No free lunch: requires hand-classified training data
- But this manual classification can be done by non-experts.
Outline

1. Classification
2. Logistic Regression
3. Logistic Regression Example
4. Motivating Naïve Bayes Example
5. Naive Bayes Definition
6. Wrapup
Generative vs. Discriminative Models

- Goal, given observation $x$, compute probability of label $y$, $p(y|x)$
- Naïve Bayes (later) uses Bayes rule to reverse conditioning
- What if we care about $p(y|x)$? We need a more general framework . . .
Generative vs. Discriminative Models

- Goal, given observation $x$, compute probability of label $y$, $p(y|x)$
- Naïve Bayes (later) uses Bayes rule to reverse conditioning
- What if we care about $p(y|x)$? We need a more general framework . . .
- That framework is called logistic regression
  - Logistic: A special mathematical function it uses
  - Regression: Combines a weight vector with observations to create an answer
  - More general cookbook for building conditional probability distributions
- Naïve Bayes (later today) is a special case of logistic regression
Logistic Regression: Definition

- Weight vector $\beta_i$
- Observations $X_i$
- “Bias” $\beta_0$ (like intercept in linear regression)

\[
P(Y = 0 | X) = \frac{1}{1 + \exp \left[ \beta_0 + \sum_i \beta_i X_i \right]} \quad (1)
\]

\[
P(Y = 1 | X) = \frac{\exp \left[ \beta_0 + \sum_i \beta_i X_i \right]}{1 + \exp \left[ \beta_0 + \sum_i \beta_i X_i \right]} \quad (2)
\]

- For shorthand, we’ll say that

\[
P(Y = 0 | X) = \sigma(- (\beta_0 + \sum_i \beta_i X_i)) \quad (3)
\]

\[
P(Y = 1 | X) = 1 - \sigma(- (\beta_0 + \sum_i \beta_i X_i)) \quad (4)
\]

- Where $\sigma(z) = \frac{1}{1 + \exp[-z]}$
**Logistic Regression**

What’s this “exp”?

### Exponential

- \( \exp[x] \) is shorthand for \( e^x \)
- \( e \) is a special number, about 2.71828
  - \( e^x \) is the limit of compound interest formula as compounds become infinitely small
  - It’s the function whose derivative is itself

### Logistic

- The “logistic” function is \( \sigma(z) = \frac{1}{1 + e^{-z}} \)
- Looks like an “S”
- Always between 0 and 1.
What’s this “exp”?  

**Exponential**  

- \( \exp[x] \) is shorthand for \( e^x \)  
- \( e \) is a special number, about 2.71828  
  - \( e^x \) is the limit of compound interest formula as compounds become infinitely small  
  - It’s the function whose derivative is itself  

**Logistic**  

- The “logistic” function is \( \sigma(z) = \frac{1}{1 + e^{-z}} \)  
- Looks like an “S”  
- Always between 0 and 1.  
  - Allows us to model probabilities  
  - Different from \texttt{linear} regression
Outline

1. Classification
2. Logistic Regression
3. Logistic Regression Example
4. Motivating Naïve Bayes Example
5. Naive Bayes Definition
6. Wrapup
### Logistic Regression Example

<table>
<thead>
<tr>
<th>feature</th>
<th>coefficient</th>
<th>weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>bias</td>
<td>$\beta_0$</td>
<td>0.1</td>
</tr>
<tr>
<td>“viagra”</td>
<td>$\beta_1$</td>
<td>2.0</td>
</tr>
<tr>
<td>“mother”</td>
<td>$\beta_2$</td>
<td>-1.0</td>
</tr>
<tr>
<td>“work”</td>
<td>$\beta_3$</td>
<td>-0.5</td>
</tr>
<tr>
<td>“nigeria”</td>
<td>$\beta_4$</td>
<td>3.0</td>
</tr>
</tbody>
</table>

- What does $Y = 1$ mean?

---

**Example 1: Empty Document?**

$X = \{\}$
### Logistic Regression Example

<table>
<thead>
<tr>
<th>feature</th>
<th>coefficient ( \beta )</th>
<th>weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>bias</td>
<td>( \beta_0 )</td>
<td>0.1</td>
</tr>
<tr>
<td>“viagra”</td>
<td>( \beta_1 )</td>
<td>2.0</td>
</tr>
<tr>
<td>“mother”</td>
<td>( \beta_2 )</td>
<td>-1.0</td>
</tr>
<tr>
<td>“work”</td>
<td>( \beta_3 )</td>
<td>-0.5</td>
</tr>
<tr>
<td>“nigeria”</td>
<td>( \beta_4 )</td>
<td>3.0</td>
</tr>
</tbody>
</table>

**Example 1: Empty Document?**

\[ X = \{ \} \]

- \( P(Y = 0) = \frac{1}{1 + \exp[0.1]} = \)
- \( P(Y = 1) = \frac{\exp[0.1]}{1 + \exp[0.1]} = \)

- What does \( Y = 1 \) mean?
### Logistic Regression Example

<table>
<thead>
<tr>
<th>feature</th>
<th>coefficient</th>
<th>weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>bias</td>
<td>$\beta_0$</td>
<td>0.1</td>
</tr>
<tr>
<td>“viagra”</td>
<td>$\beta_1$</td>
<td>2.0</td>
</tr>
<tr>
<td>“mother”</td>
<td>$\beta_2$</td>
<td>-1.0</td>
</tr>
<tr>
<td>“work”</td>
<td>$\beta_3$</td>
<td>-0.5</td>
</tr>
<tr>
<td>“nigeria”</td>
<td>$\beta_4$</td>
<td>3.0</td>
</tr>
</tbody>
</table>

- **Example 1: Empty Document?**

  $X = \{\}$

  - $P(Y = 0) = \frac{1}{1 + \exp[0.1]} = 0.48$
  - $P(Y = 1) = \frac{\exp[0.1]}{1 + \exp[0.1]} = 0.52$
  - Bias $\beta_0$ encodes the prior probability of a class

- **What does $Y = 1$ mean?**
### Logistic Regression Example

<table>
<thead>
<tr>
<th>feature</th>
<th>coefficient</th>
<th>weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>bias</td>
<td>( \beta_0 )</td>
<td>0.1</td>
</tr>
<tr>
<td>“viagra”</td>
<td>( \beta_1 )</td>
<td>2.0</td>
</tr>
<tr>
<td>“mother”</td>
<td>( \beta_2 )</td>
<td>-1.0</td>
</tr>
<tr>
<td>“work”</td>
<td>( \beta_3 )</td>
<td>-0.5</td>
</tr>
<tr>
<td>“nigeria”</td>
<td>( \beta_4 )</td>
<td>3.0</td>
</tr>
</tbody>
</table>

- What does \( Y = 1 \) mean?

**Example 2**

\[ X = \{ \text{Mother, Nigeria} \} \]
Logistic Regression Example

<table>
<thead>
<tr>
<th>feature</th>
<th>coefficient</th>
<th>weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>bias</td>
<td>$\beta_0$</td>
<td>0.1</td>
</tr>
<tr>
<td>“viagra”</td>
<td>$\beta_1$</td>
<td>2.0</td>
</tr>
<tr>
<td>“mother”</td>
<td>$\beta_2$</td>
<td>-1.0</td>
</tr>
<tr>
<td>“work”</td>
<td>$\beta_3$</td>
<td>-0.5</td>
</tr>
<tr>
<td>“nigeria”</td>
<td>$\beta_4$</td>
<td>3.0</td>
</tr>
</tbody>
</table>

- What does $Y = 1$ mean?

**Example 2**

Given $X = \{\text{Mother, Nigeria}\}$

\[
P(Y = 0) = \frac{1}{1 + \exp[0.1 - 1.0 + 3.0]} = \]

\[
P(Y = 1) = \frac{\exp[0.1 - 1.0 + 3.0]}{1 + \exp[0.1 - 1.0 + 3.0]} = \]

- Include bias, and sum the other weights
Logistic Regression Example

### Example 2

\[ X = \{ \text{Mother, Nigeria} \} \]

<table>
<thead>
<tr>
<th>feature</th>
<th>coefficient</th>
<th>weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>bias</td>
<td>( \beta_0 )</td>
<td>0.1</td>
</tr>
<tr>
<td>“viagra”</td>
<td>( \beta_1 )</td>
<td>2.0</td>
</tr>
<tr>
<td>“mother”</td>
<td>( \beta_2 )</td>
<td>-1.0</td>
</tr>
<tr>
<td>“work”</td>
<td>( \beta_3 )</td>
<td>-0.5</td>
</tr>
<tr>
<td>“nigeria”</td>
<td>( \beta_4 )</td>
<td>3.0</td>
</tr>
</tbody>
</table>

- What does \( Y = 1 \) mean?
- \( P(Y = 0) = \frac{1}{1 + \exp[0.1 - 1.0 + 3.0]} = 0.11 \)
- \( P(Y = 1) = \frac{\exp[0.1 - 1.0 + 3.0]}{1 + \exp[0.1 - 1.0 + 3.0]} = 0.88 \)
- Include bias, and sum the other weights
## Logistic Regression Example

<table>
<thead>
<tr>
<th>feature</th>
<th>coefficient</th>
<th>weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>bias</td>
<td>$\beta_0$</td>
<td>0.1</td>
</tr>
<tr>
<td>&quot;viagra&quot;</td>
<td>$\beta_1$</td>
<td>2.0</td>
</tr>
<tr>
<td>&quot;mother&quot;</td>
<td>$\beta_2$</td>
<td>-1.0</td>
</tr>
<tr>
<td>&quot;work&quot;</td>
<td>$\beta_3$</td>
<td>-0.5</td>
</tr>
<tr>
<td>&quot;nigeria&quot;</td>
<td>$\beta_4$</td>
<td>3.0</td>
</tr>
</tbody>
</table>

- What does $Y = 1$ mean?

---

**Example 3**

\[ X = \{ \text{Mother, Work, Viagra, Mother} \} \]
Logistic Regression Example

<table>
<thead>
<tr>
<th>feature</th>
<th>coefficient</th>
<th>weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>bias</td>
<td>$\beta_0$</td>
<td>0.1</td>
</tr>
<tr>
<td>“viagra”</td>
<td>$\beta_1$</td>
<td>2.0</td>
</tr>
<tr>
<td>“mother”</td>
<td>$\beta_2$</td>
<td>-1.0</td>
</tr>
<tr>
<td>“work”</td>
<td>$\beta_3$</td>
<td>-0.5</td>
</tr>
<tr>
<td>“nigeria”</td>
<td>$\beta_4$</td>
<td>3.0</td>
</tr>
</tbody>
</table>

- What does $Y = 1$ mean?

**Example 3**

$X = \{\text{Mother, Work, Viagra, Mother}\}$

- $P(Y = 0) = \frac{1}{1 + \exp [0.1 - 1.0 - 2.0 - 1.0]} =$
- $P(Y = 1) = \frac{\exp [0.1 - 1.0 - 0.5 + 2.0 - 1.0]}{1 + \exp [0.1 - 1.0 - 0.5 + 2.0 - 1.0]} =$

- Multiply feature presence by weight
Logistic Regression Example

<table>
<thead>
<tr>
<th>feature</th>
<th>coefficient</th>
<th>weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>bias</td>
<td>$\beta_0$</td>
<td>0.1</td>
</tr>
<tr>
<td>“viagra”</td>
<td>$\beta_1$</td>
<td>2.0</td>
</tr>
<tr>
<td>“mother”</td>
<td>$\beta_2$</td>
<td>-1.0</td>
</tr>
<tr>
<td>“work”</td>
<td>$\beta_3$</td>
<td>-0.5</td>
</tr>
<tr>
<td>“nigeria”</td>
<td>$\beta_4$</td>
<td>3.0</td>
</tr>
</tbody>
</table>

• What does $Y = 1$ mean?

Example 3

$X = \{\text{Mother, Work, Viagra, Mother}\}$

- $P(Y = 0) = \frac{1}{1 + \exp[0.1 - 1.0 - 0.5 + 2.0 - 1.0]} = 0.60$
- $P(Y = 1) = \frac{\exp[0.1 - 1.0 - 0.5 + 2.0 - 1.0]}{1 + \exp[0.1 - 1.0 - 0.5 + 2.0 - 1.0]} = 0.30$

• Multiply feature presence by weight
How is Logistic Regression Used?

• Given a set of weights \( \vec{\beta} \), we know how to compute the conditional likelihood \( P(y|\beta, x) \)
• Find the set of weights \( \vec{\beta} \) that maximize the conditional likelihood on training data (where \( y \) is known)
• A subset of a more general class of methods called “maximum entropy” models (next week)
• **Intuition**: higher weights mean that this feature implies that this feature is a good this is the class you want for this observation
How is Logistic Regression Used?

• Given a set of weights $\vec{\beta}$, we know how to compute the conditional likelihood $P(y|\beta,x)$

• Find the set of weights $\vec{\beta}$ that maximize the conditional likelihood on training data (where $y$ is known)

• A subset of a more general class of methods called “maximum entropy” models (next week)

• **Intuition**: higher weights mean that this feature implies that this feature is a good this is the class you want for this observation

• Naïve Bayes is a special case of logistic regression that uses Bayes rule and conditional probabilities to set these weights
Outline

1. Classification
2. Logistic Regression
3. Logistic Regression Example
4. Motivating Naïve Bayes Example
5. Naive Bayes Definition
6. Wrapup
A Classification Problem

- Suppose that I have two coins, $C_1$ and $C_2$
- Now suppose I pull a coin out of my pocket, flip it a bunch of times, record the coin and outcomes, and repeat many times:

  $C_1: 0 \ 1 \ 1 \ 1 \ 1$
  $C_1: 1 \ 1 \ 0$
  $C_2: 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1$
  $C_1: 0 \ 1$
  $C_1: 1 \ 1 \ 0 \ 1 \ 1 \ 1$
  $C_1: 0 \ 1$
  $C_2: 0 \ 0 \ 1 \ 1 \ 0 \ 1$
  $C_2: 1 \ 0 \ 0 \ 0$

- Now suppose I am given a new sequence, $0 \ 0 \ 1$; which coin is it from?
A Classification Problem

This problem has particular challenges:

- different numbers of covariates for each observation
- number of covariates can be large

However, there is some structure:

- Easy to get $P(C_1), P(C_2)$
- Also easy to get $P(X_i = 1 \mid C_1)$ and $P(X_i = 1 \mid C_2)$
- By conditional independence,

$$P(X = 010 \mid C_1) = P(X_1 = 0 \mid C_1)P(X_2 = 1 \mid C_1)P(X_2 = 0 \mid C_1)$$

- Can we use these to get $P(C_1 \mid X = 001)$?
A Classification Problem

This problem has particular challenges:

• different numbers of covariates for each observation
• number of covariates can be large

However, there is some structure:

• Easy to get $P(C_1) = 4/7$, $P(C_2) = 3/7$
• Also easy to get $P(X_i = 1 | C_1)$ and $P(X_i = 1 | C_2)$
• By conditional independence,

\[
P(X = 010 | C_1) = P(X_1 = 0 | C_1) P(X_2 = 1 | C_1) P(X_2 = 0 | C_1)
\]
• Can we use these to get $P(C_1 | X = 001)$?
A Classification Problem

This problem has particular challenges:

• different numbers of covariates for each observation
• number of covariates can be large

However, there is some structure:

• Easy to get $P(C_1) = 4/7$, $P(C_2) = 3/7$
• Also easy to get $P(X_i = 1 \mid C_1) = 12/16$ and $P(X_i = 1 \mid C_2) = 6/18$
• By conditional independence,

$$P(X = 0 1 0 \mid C_1) = P(X_1 = 0 \mid C_1) P(X_2 = 1 \mid C_1) P(X_2 = 0 \mid C_1)$$

• Can we use these to get $P(C_1 \mid X = 0 0 1)$?
A Classification Problem

Summary: have $P(data | class)$, want $P(class | data)$

Solution: Bayes’ rule!

$$P(class | data) = \frac{P(data | class)P(class)}{P(data)}$$

$$= \frac{P(data | class)P(class)}{\sum_{class=1}^{C} P(data | class)P(class)}$$

To compute, we need to estimate $P(data | class)$, $P(class)$ for all classes
Naive Bayes Classifier

This works because the coin flips are independent given the coin parameter. What about this case:

- want to identify the type of fruit given a set of features: color, shape and size
- color: red, green, yellow or orange (discrete)
- shape: round, oval or long+skinny (discrete)
- size: diameter in inches (continuous)
Naive Bayes Classifier

Conditioned on type of fruit, these features are not necessarily independent:

\[ P(\text{green} | \text{size} < 2, \text{apple}) > P(\text{green} | \text{apple}) \]
Naive Bayes Classifier

Using chain rule,

\[
P(apple|\text{green, round, size}=2) = \frac{P(\text{green, round, size}=2|apple)P(apple)}{\sum_{\text{fruits}} P(\text{green, round, size}=2|\text{fruit } j)P(\text{fruit } j)} \]

\[
= P(\text{green|round, size}=2, apple)P(\text{round|size}=2, apple) \times P(\text{size}=2|\text{apple})P(\text{apple})
\]

But computing conditional probabilities is hard! There are many combinations of \((\text{color, shape, size})\) for each fruit.
Naive Bayes Classifier

Idea: assume conditional independence for all features given class,

\[ P(green | round, size = 2, apple) = P(green | apple) \]
\[ P(round | green, size = 2, apple) = P(round | apple) \]
\[ P(size = 2 | green, round, apple) = P(size = 2 | apple) \]
Outline

1. Classification
2. Logistic Regression
3. Logistic Regression Example
4. Motivating Naïve Bayes Example
5. Naive Bayes Definition
6. Wrapup
The Naive Bayes classifier

- The Naive Bayes classifier is a probabilistic classifier.
- We compute the probability of a document $d$ being in a class $c$ as follows:

$$ P(c|d) \propto P(c) \prod_{1 \leq i \leq n_d} P(w_i|c) $$

$P(c|d)$ is the probability of a document $d$ being in class $c$.
$P(c)$ is the prior probability of $c$.
$P(w_i|c)$ is the conditional probability of term $w_i$ occurring in a document of class $c$.
The Naive Bayes classifier

- The Naive Bayes classifier is a probabilistic classifier.
- We compute the probability of a document \(d\) being in a class \(c\) as follows:

\[
P(c|d) \propto P(c) \prod_{1 \leq i \leq n_d} P(w_i|c)
\]

- \(n_d\) is the length of the document (number of tokens).
- \(P(w_i|c)\) is the conditional probability of term \(w_i\) occurring in a document of class \(c\).
- \(P(w_i|c)\) as a measure of how much evidence \(w_i\) contributes that \(c\) is the correct class.
- \(P(c)\) is the prior probability of \(c\).
- If a document's terms do not provide clear evidence for one class vs. another, we choose the \(c\) with higher \(P(c)\).
The Naive Bayes classifier

- The Naive Bayes classifier is a probabilistic classifier.
- We compute the probability of a document $d$ being in a class $c$ as follows:

$$P(c|d) \propto P(c) \prod_{1 \leq i \leq n_d} P(w_i|c)$$

- $n_d$ is the length of the document. (number of tokens)
- $P(w_i|c)$ is the conditional probability of term $w_i$ occurring in a document of class $c$
- $P(w_i|c)$ as a measure of how much evidence $w_i$ contributes that $c$ is the correct class.
- $P(c)$ is the prior probability of $c$.
- If a document’s terms do not provide clear evidence for one class vs. another, we choose the $c$ with higher $P(c)$. 
Maximum a posteriori class

- Our goal is to find the “best” class.
- The best class in Naive Bayes classification is the most likely or *maximum a posteriori (MAP) class* $c_{\text{map}}$:

$$c_{\text{map}} = \arg \max_{c_j \in \mathcal{C}} \hat{P}(c_j|d) = \arg \max_{c_j \in \mathcal{C}} \hat{P}(c_j) \prod_{1 \leq i \leq n_d} \hat{P}(w_i|c_j)$$

- We write $\hat{P}$ for $P$ since these values are *estimates* from the training set.
Naive Bayes Classifier

Why conditional independence?

- estimating multivariate functions (like $P(X_1, \ldots, X_m \mid Y)$) is mathematically hard, while estimating univariate ones is easier (like $P(X_i \mid Y)$)
- need less data to fit univariate functions well
- univariate estimators differ much less than multivariate estimator (low variance)
- ... but they may end up finding the wrong values (more bias)
Naïve Bayes conditional independence assumption

To reduce the number of parameters to a manageable size, recall the *Naive Bayes conditional independence assumption*:

$$ P(d|c_j) = P(w_1, \ldots, w_{n_d}|c_j) = \prod_{1 \leq i \leq n_d} P(X_i = w_i|c_j) $$

We assume that the probability of observing the conjunction of attributes is equal to the product of the individual probabilities $P(X_i = w_i|c_j)$.

Our estimates for these priors and conditional probabilities:

$$ \hat{P}(c_j) = \frac{N_c + 1}{N + |C|} $$

and

$$ \hat{P}(w|c) = \frac{T_{cw} + 1}{(\sum_{w' \in V} T_{cw'}) + |V|} $$
Implementation Detail: Taking the log

- Multiplying lots of small probabilities can result in floating point underflow.
- From last time $\lg$ is logarithm base 2; $\ln$ is logarithm base $e$.

\[
\begin{align*}
\lg x &= a \iff 2^a = x \\
\ln x &= a \iff e^a = x
\end{align*}
\] (5)

- Since $\ln(xy) = \ln(x) + \ln(y)$, we can sum log probabilities instead of multiplying probabilities.
- Since $\ln$ is a monotonic function, the class with the highest score does not change.
- So what we usually compute in practice is:

\[
c\;\text{map} = \arg\max_{c_j \in \mathbb{C}} \left[ \hat{P}(c_j) \prod_{1 \leq i \leq n_d} \hat{P}(w_i|c_j) \right]
\]

\[
\arg\max_{c_j \in \mathbb{C}} \left[ \ln \hat{P}(c_j) + \sum_{1 \leq i \leq n_d} \ln \hat{P}(w_i|c_j) \right]
\]
Naive Bayes Definition

Implementation Detail: Taking the log

- Multiplying lots of small probabilities can result in floating point underflow.
- From last time \( \text{lg} \) is logarithm base 2; \( \ln \) is logarithm base \( e \).

\[
\text{lg} \, x = a \iff 2^a = x \quad \text{ln} \, x = a \iff e^a = x \quad (5)
\]

- Since \( \ln(xy) = \ln(x) + \ln(y) \), we can sum log probabilities instead of multiplying probabilities.
- Since \( \ln \) is a monotonic function, the class with the highest score does not change.
- So what we usually compute in practice is:

\[
c \text{ map } = \arg \max_{c_j \in C} \left[ \hat{P}(c_j) \prod_{1 \leq i \leq n_d} \hat{P}(w_i|c_j) \right]
\]

\[
\arg \max_{c_j \in C} \left[ \ln \hat{P}(c_j) + \sum_{1 \leq i \leq n_d} \ln \hat{P}(w_i|c_j) \right]
\]
Implementation Detail: Taking the log

- Multiplying lots of small probabilities can result in floating point underflow.
- From last time \( \lg \) is logarithm base 2; \( \ln \) is logarithm base \( e \).

\[
\begin{align*}
\lg x &= a \iff 2^a = x \\
\ln x &= a \iff e^a = x \quad (5)
\end{align*}
\]

- Since \( \ln(xy) = \ln(x) + \ln(y) \), we can sum log probabilities instead of multiplying probabilities.
- Since \( \ln \) is a monotonic function, the class with the highest score does not change.
- So what we usually compute in practice is:

\[
c_{\text{map}} = \arg\max_{c_j \in \mathbb{C}} \left[ \hat{P}(c_j) \prod_{1 \leq i \leq n_d} \hat{P}(w_i | c_j) \right] \\
= \arg\max_{c_j \in \mathbb{C}} \left[ \ln \hat{P}(c_j) + \sum_{1 \leq i \leq n_d} \ln \hat{P}(w_i | c_j) \right]
\]
Outline

1. Classification
2. Logistic Regression
3. Logistic Regression Example
4. Motivating Naïve Bayes Example
5. Naive Bayes Definition
6. Wrapup
Equivalence of Naïve Bayes and Logistic Regression

Consider Naïve Bayes and logistic regression with two classes: (+) and (-).

### Naïve Bayes

\[
\hat{P}(c_+) \prod_i \hat{P}(w_i|c_+)
\]

\[
\hat{P}(c_-) \prod_i \hat{P}(w_i|c_-)
\]

### Logistic Regression

\[
\sigma \left( -\beta_0 - \sum_i \beta_i X_i \right) = \frac{1}{1 + \exp \left( \beta_0 + \sum_i \beta_i X_i \right)}
\]

\[
1 - \sigma \left( -\beta_0 - \sum_i \beta_i X_i \right) = \frac{\exp \left( \beta_0 + \sum_i \beta_i X_i \right)}{1 + \exp \left( \beta_0 + \sum_i \beta_i X_i \right)}
\]

- These are actually the same if
  \[
  w_0 = \sigma \left( \ln \left( \frac{p(c_+)}{1-p(c_+)} \right) + \sum_j \ln \left( \frac{1-P(w_j|c_+)}{1-P(w_j|c_-)} \right) \right)
  \]
- and
  \[
  w_j = \ln \left( \frac{P(w_j|c_+)(1-P(w_j|c_-))}{P(w_j|c_-)(1-P(w_j|c_+))} \right)
  \]
Contrasting Naïve Bayes and Logistic Regression

- Naïve Bayes easier
- Naïve Bayes better on smaller datasets
- Logistic regression better on medium-sized datasets
- On huge datasets, it doesn’t really matter (data always win)
  - Optional reading by Ng and Jordan has proofs and experiments
- Logistic regression allows arbitrary features (biggest difference!)
Contrasting Naïve Bayes and Logistic Regression

- Naïve Bayes easier
- Naïve Bayes better on smaller datasets
- Logistic regression better on medium-sized datasets
- On huge datasets, it doesn’t really matter (data always win)
  - Optional reading by Ng and Jordan has proofs and experiments
- Logistic regression allows arbitrary features (biggest difference!)
- Don’t need to memorize (or work through) previous slide—just understand that naïve Bayes is a special case of logistic regression
In class
In class
In class
In class
Next time . . .

- Maximum Entropy: Mathematical foundations to logistic regression
- How to learn the best setting of weights
- Extracting features from words