Supervised Learning: Naïve Bayes

• Basic concept:
  • Problem:
    • Given data \( D = \{ (\bar{x}_i, y_i) \} \) for \( 1 \leq i \leq N \)
    • Label new item \( \bar{x}_{\text{test}} \)
  • Solution:
    • Assign label
      \[
      \arg\max_{c \in C} P(c | \bar{x}_{\text{test},1} = v_{1,l_1}, \bar{x}_{\text{test},2} = v_{2,l_2}, \ldots, \bar{x}_{\text{test},n} = v_{n,l_{k1}})
      \]
      [this says: whichever \( c \) is most probable for \( \bar{x}_{\text{test}} \)]
    • When clear from context, will write:
      \[
      \arg\max_{c \in C} P(c | \bar{x}_{\text{test},1}, \ldots, \bar{x}_{\text{test},n})
      \]
Supervised Learning: Naïve Bayes

Bayes Rule:

\[ P(A|B) = \frac{P(B|A)P(A)}{P(B)} \]
Supervised Learning: Naïve Bayes

Bayes Rule:

\[ P(A|B) = \frac{P(B|A)P(A)}{P(B)} \]

so

\[ \arg\max_{c \in C} P(c|\tilde{x}_{test,1}, \ldots, \tilde{x}_{test,n}) = \arg\max_{c \in C} \frac{P(\tilde{x}_{test,1}, \ldots, \tilde{x}_{test,n}|c)P(c)}{P(\tilde{x}_{test,1}, \ldots, \tilde{x}_{test,n})} \]
Supervised Learning: Naïve Bayes

Bayes Rule:

\[ P(A|B) = \frac{P(B|A)P(A)}{P(B)} \]

so

\[ \arg\max_{c \in C} P(c|\tilde{x}_{test,1}, \ldots, \tilde{x}_{test,n}) = \arg\max_{c \in C} \frac{P(\tilde{x}_{test,1}, \ldots, \tilde{x}_{test,n}|c)P(c)}{P(\tilde{x}_{test,1}, \ldots, \tilde{x}_{test,n})} \]

\[ = \arg\max_{c \in C} P(\tilde{x}_{test,1}, \ldots, \tilde{x}_{test,n}|c)P(c) \]
Supervised Learning: Naïve Bayes

Bayes Rule:

\[ P(A|B) = \frac{P(B|A)P(A)}{P(B)} \]

so

\[
\arg\max_{c \in C} P(c|\tilde{x}_{test,1}, \ldots, \tilde{x}_{test,n}) = \arg\max_{c \in C} \frac{P(\tilde{x}_{test,1}, \ldots, \tilde{x}_{test,n}|c)P(c)}{P(\tilde{x}_{test,1}, \ldots, \tilde{x}_{test,n})}
\]

\[ = \arg\max_{c \in C} P(\tilde{x}_{test,1}, \ldots, \tilde{x}_{test,n}|c)P(c) \]

The denominator \( P(\tilde{x}_{test,1}, \ldots, \tilde{x}_{test,n}) \) is the same constant for all \( c \) and thus doesn’t change the argmax outcome
Supervised Learning: Naïve Bayes

Bayes Rule:

\[ P(A|B) = \frac{P(B|A)P(A)}{P(B)} \]

so

\[ \arg\max_{c \in C} P(c|\tilde{x}_{test,1}, \ldots, \tilde{x}_{test,n}) = \arg\max_{c \in C} \frac{P(\tilde{x}_{test,1}, \ldots, \tilde{x}_{test,n}|c)P(c)}{P(\tilde{x}_{test,1}, \ldots, \tilde{x}_{test,n})} \]

\[ = \arg\max_{c \in C} P(\tilde{x}_{test,1}, \ldots, \tilde{x}_{test,n}|c)P(c) \]
Supervised Learning: Naïve Bayes

Bayes Rule:

\[ P(A|B) = \frac{P(B|A)P(A)}{P(B)} \]

so

\[ \arg\max_{c \in C} P(c|\tilde{x}_{test,1}, \ldots, \tilde{x}_{test,n}) = \arg\max_{c \in C} \frac{P(\tilde{x}_{test,1}, \ldots, \tilde{x}_{test,n}|c)P(c)}{P(\tilde{x}_{test,1}, \ldots, \tilde{x}_{test,n})} \]

\[ = \arg\max_{c \in C} P(\tilde{x}_{test,1}, \ldots, \tilde{x}_{test,n}|c)P(c) \]

\[ = \arg\max_{c \in C} P(c) P(\tilde{x}_{test,1}, \ldots, \tilde{x}_{test,n}|c) \]

Commonly written with \( P(c) \) first
Supervised Learning: Naïve Bayes

\[
\arg\max_{c \in C} P(c | \tilde{x}_{test,1}, \ldots, \tilde{x}_{test,n}) = \arg\max_{c \in C} P(c) P(\tilde{x}_{test,1}, \ldots, \tilde{x}_{test,n} | c)
\]

\[
\approx \arg\max_{c \in C} P(c) \prod_{j=1}^{n} P(\tilde{x}_{test,j} | c)
\]

Simplification: “Conditional Independence”

(Example airline travel: P(person_1 is late, person_2 is late | day)
Supervised Learning: Naïve Bayes

Assign label $c = \arg\max_{c \in C} P(c) \prod_{j=1}^{n} P(\tilde{x}_{test,j} | c)$
Supervised Learning: Naïve Bayes

Assign label $c = \arg\max_{c \in C} P(c) \prod_{j=1}^{n} P(\tilde{x}_{test,j} | c)$

Estimated from the data
Supervised Learning: Naïve Bayes

Notation:

\[ N = |D| = \text{total number of examples} \]
\[ N_i = |\{<\tilde{x}, y> \in D \mid y=c_i}\}| = \text{number of examples of class } c_i \]
Supervised Learning: Naïve Bayes

Notation:

\[ N = |D| = \text{total number of examples} \]
\[ N_i = |\{<\hat{x}, y> \in D | y=c_i}\}| = \text{number of examples of class } c_i \]

Assign label \( c = \arg\max_{c \in C} P(c) \prod_{j=1}^{n} P(\hat{x}_{test,j}|c) \)
Supervised Learning: Naïve Bayes

Notation:

\[ N = |D| = \text{total number of examples} \]
\[ N_i = |\{<\tilde{x},y> \in D \mid y=c_i}\}| = \text{number of examples of class } c_i \]

Assign label \( c = \arg\max_{c \in C} P(c) \prod_{j=1}^{n} P(\tilde{x}_{test,j} | c) \)
Supervised Learning: Naïve Bayes

Notation:

\[ N = |D| = \text{total number of examples} \]

\[ N_l = \left| \{<\tilde{x}, y> \in D \mid y = c_l \} \right| = \text{number of examples of class } c_l \]

Assign label \( c = \arg\max_{c \in C} \left[ P(c) \prod_{j=1}^{n} P(\tilde{x}_{test,j} \mid c) \right] \)

Estimate \( P(c_l) = \frac{N_l}{N} \)
Supervised Learning: Naïve Bayes

Notation:

\[ N = |D| = \text{total number of examples} \]
\[ N_i = |\{<\hat{x},y> \in D \mid y=c_i}\}| = \text{number of examples of class } c_i \]

Assign label \( c = \arg\max_{c \in C} P(c) \prod_{j=1}^{n} P(\hat{x}_{test,j} \mid c) \)
Supervised Learning: Naïve Bayes

Notation:

\[ N = |D| = \text{total number of examples} \]

\[ N_i = |\{<\tilde{x},y> \in D \mid y=c_i}\}| = \text{number of examples of class } c_i \]

\[ n_{jkl} = |\{<\tilde{x},y> \in D \mid \tilde{x}_j=v_k \text{ and } y=c_i}\}| \]

\[ = \text{number of examples with } \tilde{x}_j = v_k \text{ and } y=c_i \]
Supervised Learning: Naïve Bayes

Notation:

\[ N = |D| = \text{total number of examples} \]
\[ N_l = |\{<\tilde{x}, y> \in D \mid y = c_l}\}| = \text{number of examples of class } c_l \]
\[ n_{jkl} = |\{<\tilde{x}, y> \in D \mid \tilde{x}_j = v_k \text{ and } y = c_l\}| \]
\[ = \text{number of examples with } \tilde{x}_j = v_k \text{ and } y = c_l \]

Assign label \( c = \arg \max_{c \in C} P(c) \prod_{j=1}^{m} P(\tilde{x}_{\text{test}, j} | c) \)
Supervised Learning: Naïve Bayes

Notation:

\[ N = |D| = \text{total number of examples} \]
\[ N_l = |\{<\bar{x},y> \in D \mid y=c_l\}| = \text{number of examples of class } c_l \]
\[ n_{jkl} = |\{<\bar{x},y> \in D \mid \bar{x}_j = v_k \text{ and } y = c_l\}| = \text{number of examples with } \bar{x}_j = v_k \text{ and } y = c_l \]

Assign label \( c = \arg\max_{c \in C} P(c) \prod_{j=1}^{n} P(\bar{x}_{test,j} | c) \)

Estimate \( P(\bar{x}_{test,j} = v_k | c_l) = \frac{n_{jkl}}{N_l} \)
Supervised Learning: Naïve Bayes

Assign label $c$ for which

$$\arg\max_{c \in C} P(c | \tilde{x}_{\text{test},1}, \ldots, \tilde{x}_{\text{test},n})$$

$$= \arg\max_{c \in C} P(c) \prod_{j=1}^{n} P(\tilde{x}_{\text{test},j} | c)$$

$$= \arg\max_{c_l \in C} \frac{N_l}{N} \prod_{j=1}^{n} \frac{n_{jkj}}{N_l}$$
Supervised Learning: Naïve Bayes

Assign label c for which

$$\arg\max_{c \in C} P(c | \tilde{x}_{test,1}, \ldots, \tilde{x}_{test,n})$$

$$= \arg\max_{c \in C} P(c) \prod_{j=1}^{n} P(\tilde{x}_{test,j} | c)$$

$$= \arg\max_{c_l \in C} \frac{N_l}{N} \prod_{j=1}^{n} \frac{n_{j,k,l}}{N_l}$$

$k_j$: the value in $V_j$ that the test example takes on
Supervised Learning: Naïve Bayes

Assign label $c = \arg\max_{c_l \in C} \frac{N_l}{N} \prod_{j=1}^{n} \frac{n_{jkj\ell}}{N_l}$

Implementation trick:
Multiplying a lot of probabilities together gives tiny numbers
“Underflow”
Supervised Learning: Naïve Bayes

Assign label $c = \arg \max_{c_l \in C} \frac{N_l}{N} \prod_{j=1}^{n} \frac{n_{j k j l}}{N_l}$

Implementation trick:
Multiplying a lot of probabilities together gives tiny numbers
“Underflow”

Notice for $a, b > 0$
$a > b$ if and only if $\log a > \log b$
Supervised Learning: Naïve Bayes

Assign label \( c = \arg\max_{c_l \in C} \)

\[
\left\{ \frac{N_l}{N} \prod_{j=1}^{n} \frac{n_{jkj}}{N_l} \right\}
\]

Taking the log doesn’t change the argmax outcome
Supervised Learning: Naïve Bayes

Assign label \( c = \arg\max_{c_l \in C} \) 

\[
\begin{align*}
\text{Taking the log doesn’t} & \quad \text{change the argmax outcome} \\
= \arg\max_{c_l \in C} \log \left[ \frac{N_l}{N} \prod_{j=1}^{n} \frac{n_{jkj}}{N_l} \right] + \log \prod_{j=1}^{n} \frac{n_{jkj}}{N_l} \\
= \arg\max_{c_l \in C} \left( \sum_{j=1}^{n} \log n_{jkj} \right) - (n - 1) \log N_l \\
\end{align*}
\]

Algebra
Supervised Learning: Naïve Bayes

Assign label \( c = \arg\max_{c_l \in C} \frac{N_l}{N} \prod_{j=1}^{n} \frac{n_{j \cdot k_l}}{N_l} \)

\[
= \arg\max_{c_l \in C} \log \left( \frac{N_l}{N} \right) + \log \prod_{j=1}^{n} \frac{n_{j \cdot k_l}}{N_l}
\]

\[
= \arg\max_{c_l \in C} \left( \sum_{j=1}^{n} \log n_{j \cdot k_l} \right) - (n - 1) \log N_l
\]

Taking the log doesn’t change the argmax outcome.

Gives the same outcome, but avoids underflow.

Algebra
Supervised Learning: Naïve Bayes

Special case: binary attributes

Imagine $V_j = \{0,1\}$ for each feature $j$
So for each feature $j$ either $x_j = 0$ or $x_j = 1$
Supervised Learning: Naïve Bayes

Special case: binary attributes

We estimated \( P(\bar{x}_j = v_k \mid c_l) \) with \( \frac{n_{jk_l}}{N_l} \)

We can rewrite this as \( \left( \frac{n_{j1l}}{N_l} \right)^{x_j} \left( \frac{n_{j0l}}{N_l} \right)^{1-x_j} \)  

Uses \( x_j \) as a sort of case statement
Supervised Learning: Naïve Bayes

Special case: binary attributes

We estimated \( P(\bar{x}_j = v_k \mid c_l) \) with \( \frac{n_{jkjl}}{N_l} \)

We can rewrite this as \( \left( \frac{n_{j1l}}{N_l} \right)^{x_j} \left( \frac{n_{j0l}}{N_l} \right)^{1-x_j} \)

This can in turn be written as \( \left( \frac{n_{j1l}}{N_l} \right)^{x_j} \left( 1 - \frac{n_{j1l}}{N_l} \right)^{1-x_j} \) Algebra, using \( n_{j0l} + n_{j1l} = N_l \)
Supervised Learning: Naïve Bayes

Special case: binary attributes

Assign label \( c = \arg\max_{c_l \in C} \left[ \frac{N_l}{N} \prod_{j=1}^{n} \frac{n_{jk \cdot l}}{N_l} \right] \)

\[
= \arg\max_{c_l \in C} \left[ \frac{N_l}{N} \prod_{j=1}^{n} \left( \frac{n_{j1 \cdot l}}{N_l} \right)^{x_j} \left( 1 - \frac{n_{j1 \cdot l}}{N_l} \right)^{1-x_j} \right]
\]
Supervised Learning: Naïve Bayes

Special case: binary attributes

Assign label $c = \operatorname{argmax}_{c_l \in C} \left[ \frac{N_l}{N} \prod_{j=1}^{n} \frac{n_{jkjl}}{N_l} \right]$  

$= \operatorname{argmax}_{c_l \in C} \left[ \frac{N_l}{N} \prod_{j=1}^{n} \left( \frac{n_{j1l}}{N_l} \right)^{x_j} \left( 1 - \frac{n_{j1l}}{N_l} \right)^{1-x_j} \right]$  

Looks very different, does the same thing