Security Analytics

Topic 5: Probabilistic Classification Models: Naïve Bayes

Purdue University
Prof. Ninghui Li
Based on slides by Prof. Jenifer Neville and Chris Clifton
Readings

• Principle of Data Mining
  – Chapter 10: Predictive Modeling for Classification
    • 10.8 The Naïve Bayes Model

• From Speech and Language Processing. Daniel Jurafsky & James H. Martin
  – Chapter 6: Naive Bayes and Sentiment Classification
The Classification Problem

• Given input $\mathbf{x}$, the goal is to predict $y$, which is a categorical variable
  – $y$ is called the class label
  – $\mathbf{x}$ is the feature vector

• Example:
  – $\mathbf{x}$: monthly income and bank saving amount; $y$: risky or not risky
  – $\mathbf{x}$: bag-of-words representation of an email; $y$: spam or not spam
Precision and Recall

• Given a dataset, we train a classifier that gets 99% accuracy

• Did we do a good job?

• Build a classifier for brain tumor:
  • 99.9% of brain scans do not show signs of tumor
  • Did we do a good job?

• By simply saying “NO” to all examples we reduce the error by a factor of 10!
  • Clearly Accuracy is not the best way to evaluate the learning system when the data is heavily skewed!

• Intuition: we need a measure that captures the class we care about! (rare)
Precision and Recall

The learner can make two kinds of mistakes:

- **False Positive**
- **False Negative**

<table>
<thead>
<tr>
<th>Predicted</th>
<th>True Label</th>
<th>False Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>True Positive</td>
<td>False Positive</td>
</tr>
<tr>
<td>0</td>
<td>False Negative</td>
<td>True Negative</td>
</tr>
</tbody>
</table>

**Precision:**

\[
\text{Precision} = \frac{\text{True Pos}}{\text{Predicted Pos}} = \frac{\text{True Pos}}{\text{True Pos} + \text{False Pos}}
\]

“When we predicted the rare class, how often are we right?”

**Recall**

\[
\text{Recall} = \frac{\text{True Pos}}{\text{Actual Pos}} = \frac{\text{True Pos}}{\text{True Pos} + \text{False Neg}}
\]

“Out of all the instances of the rare class, how many did we catch?”
Precision and Recall

- Precision and Recall give us two reference points to compare learning performance

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>Algorithm 1</td>
<td>0.5</td>
<td>0.4</td>
</tr>
<tr>
<td>Algorithm 2</td>
<td>0.7</td>
<td>0.1</td>
</tr>
<tr>
<td>Algorithm 3</td>
<td>0.02</td>
<td>1</td>
</tr>
</tbody>
</table>

- Which algorithm is better? We need a single score

- Properties of f-score:
  - Ranges between 0-1
  - Prefers precision and recall with similar values

- Option 1: Average
  \[
  \frac{P + R}{2}
  \]

- Option 2: F-Score
  \[
  \frac{2PR}{P + R}
  \]
NAÏVE BAYES CLASSIFIER
Example

• Example: Play Tennis

*PlayTennis: training examples*

<table>
<thead>
<tr>
<th>Day</th>
<th>Outlook</th>
<th>Temperature</th>
<th>Humidity</th>
<th>Wind</th>
<th>PlayTennis</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1</td>
<td>Sunny</td>
<td>Hot</td>
<td>High</td>
<td>Weak</td>
<td>No</td>
</tr>
<tr>
<td>D2</td>
<td>Sunny</td>
<td>Hot</td>
<td>High</td>
<td>Strong</td>
<td>No</td>
</tr>
<tr>
<td>D3</td>
<td>Overcast</td>
<td>Hot</td>
<td>High</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D4</td>
<td>Rain</td>
<td>Mild</td>
<td>High</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D5</td>
<td>Rain</td>
<td>Cool</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D6</td>
<td>Rain</td>
<td>Cool</td>
<td>Normal</td>
<td>Strong</td>
<td>No</td>
</tr>
<tr>
<td>D7</td>
<td>Overcast</td>
<td>Cool</td>
<td>Normal</td>
<td>Strong</td>
<td>Yes</td>
</tr>
<tr>
<td>D8</td>
<td>Sunny</td>
<td>Mild</td>
<td>High</td>
<td>Weak</td>
<td>No</td>
</tr>
<tr>
<td>D9</td>
<td>Sunny</td>
<td>Cool</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D10</td>
<td>Rain</td>
<td>Mild</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D11</td>
<td>Sunny</td>
<td>Mild</td>
<td>Normal</td>
<td>Strong</td>
<td>Yes</td>
</tr>
<tr>
<td>D12</td>
<td>Overcast</td>
<td>Mild</td>
<td>High</td>
<td>Strong</td>
<td>Yes</td>
</tr>
<tr>
<td>D13</td>
<td>Overcast</td>
<td>Hot</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D14</td>
<td>Rain</td>
<td>Mild</td>
<td>High</td>
<td>Strong</td>
<td>No</td>
</tr>
</tbody>
</table>
Probabilistic Classification

- Establishing a probabilistic model for classification
  - **Discriminative model**

  \[
P(c \mid x) \quad c = c_1, \ldots, c_L, \quad x = (x_1, \ldots, x_n)
\]

- To train a discriminative classifier regardless of its probabilistic or non-probabilistic nature, all training examples of different classes must be jointly used to build up a single discriminative classifier.

- Output \( L \) probabilities for \( L \) class labels in a probabilistic classifier while a single label is achieved by a non-probabilistic classifier.

- Example: Logistic Regression, SVM, etc.
Probabilistic Classification

Establishing a probabilistic model for classification (cont.)

- **Generative model (must be probabilistic)**

\[
P(x | c) \quad c = c_1, \cdots, c_L, \quad x = (x_1, \cdots, x_n)
\]

- \(L\) probabilistic models have to be trained independently
- Each is trained on only the examples of the same label
- Output \(L\) probabilities for a given input with \(L\) models
- “Generative” means that such a model produces data subject to the distribution via sampling.
Bayes rule for probabilistic classifier

- The learner considers a set of candidate labels, and attempts to find the most probable one \( y \in Y \), given the observed data.

- Such maximally probable assignment is called maximum a posteriori assignment (MAP); Bayes theorem is used to compute it:

\[
y_{\text{MAP}} = \arg\max_{y \in Y} P(y|x) = \arg\max_{y \in Y} P(x|y) P(y)/P(x)
\]

\[
= \arg\max_{y \in Y} P(x|y) P(y)
\]

Since \( P(x) \) is the same for all \( y \in Y \)
Bayes Classifier

**Maximum A Posterior (MAP) classification rule**

For an input $x$, find the largest one from $L$ probabilities output by a discriminative probabilistic classifier

$$P(c_1 | x), ..., P(c_L | x).$$

Assign $x$ to label $c^*$ if $P(c^* | x)$ is the largest.

- **Generative classification with the MAP rule**
  - Apply Bayesian rule to convert them into posterior probabilities

$$P(c_i | x) = \frac{P(x | c_i)P(c_i)}{P(x)} \propto P(x | c_i)P(c_i)$$

for $i = 1, 2, ..., L$

- Then apply the MAP rule to assign a label

Common factor for all $L$ probabilities
Naïve Bayes

- **Bayes classification**

\[ P(c \mid x) \propto P(x \mid c)P(c) = P(x_1, \ldots, x_n \mid c)P(c) \text{ for } c = c_1, \ldots, c_L. \]

Difficulty: learning the joint probability \( P(x_1, \ldots, x_n \mid c) \) is infeasible!

- **Naïve Bayes classification**
  - Assume all input features are class conditionally independent!

\[
P(x_1, x_2, \ldots, x_n \mid c) = P(x_1 \mid x_2, \ldots, x_n, c)P(x_2, \ldots, x_n \mid c)
\]

Applying the independence assumption

\[
= P(x_1 \mid c)P(x_2, \ldots, x_n \mid c)
\]

\[
= P(x_1 \mid c)P(x_2 \mid c) \cdots P(x_n \mid c)
\]

- Apply the MAP classification rule: assign \( x' = (a_1, a_2, \ldots, a_n) \) to \( c^* \)

\[
[P(a_1 \mid c^*) \cdots P(a_n \mid c^*)]P(c^*) > [P(a_1 \mid c) \cdots P(a_n \mid c)]P(c), \quad c \neq c^*, c = c_1, \ldots, c_L
\]

\[
\text{estimate of } P(a_1, \ldots, a_n \mid c^*) \quad \text{estimate of } P(a_1, \ldots, a_n \mid c)
\]
Naïve Bayes

- Algorithm: Discrete-Valued Features
  - **Learning Phase**: Given a training set $S$ of $F$ features and $L$ classes,
    
    For each target value of $c_i$ ($c_i = c_1, \ldots, c_L$)
    
    $$\hat{P}(c_i) \leftarrow \text{estimate } P(c_i) \text{ with examples in } S;$$
    
    For every feature value $x_{jk}$ of each feature $x_j$ ($j = 1, \ldots, F; k = 1, \ldots, N_j$)
    
    $$\hat{P}(x_j = x_{jk} \mid c_i) \leftarrow \text{estimate } P(x_{jk} \mid c_i) \text{ with examples in } S;$$
    
    **Output**: $F \times L$ conditional probabilistic (generative) models
  - **Test Phase**: Given an unknown instance $x' = (a'_1, \ldots, a'_n)$
    
    “Look up tables” to assign the label $c^*$ to $X'$ if
    
    $$\hat{P}(a'_1 \mid c^*) \cdots \hat{P}(a'_n \mid c^*)\hat{P}(c^*) > \hat{P}(a'_1 \mid c_1) \cdots \hat{P}(a'_n \mid c_i)\hat{P}(c_i), \quad c_i \neq c^*, c_i = c_1, \ldots, c_L$$
**Example**

- **Example: Play Tennis**

*PlayTennis: training examples*

<table>
<thead>
<tr>
<th>Day</th>
<th>Outlook</th>
<th>Temperature</th>
<th>Humidity</th>
<th>Wind</th>
<th>PlayTennis</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1</td>
<td>Sunny</td>
<td>Hot</td>
<td>High</td>
<td>Weak</td>
<td>No</td>
</tr>
<tr>
<td>D2</td>
<td>Sunny</td>
<td>Hot</td>
<td>High</td>
<td>Strong</td>
<td>No</td>
</tr>
<tr>
<td>D3</td>
<td>Overcast</td>
<td>Hot</td>
<td>High</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D4</td>
<td>Rain</td>
<td>Mild</td>
<td>High</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D5</td>
<td>Rain</td>
<td>Cool</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D6</td>
<td>Rain</td>
<td>Cool</td>
<td>Normal</td>
<td>Strong</td>
<td>No</td>
</tr>
<tr>
<td>D7</td>
<td>Overcast</td>
<td>Cool</td>
<td>Normal</td>
<td>Strong</td>
<td>Yes</td>
</tr>
<tr>
<td>D8</td>
<td>Sunny</td>
<td>Mild</td>
<td>High</td>
<td>Weak</td>
<td>No</td>
</tr>
<tr>
<td>D9</td>
<td>Sunny</td>
<td>Cool</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D10</td>
<td>Rain</td>
<td>Mild</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D11</td>
<td>Sunny</td>
<td>Mild</td>
<td>Normal</td>
<td>Strong</td>
<td>Yes</td>
</tr>
<tr>
<td>D12</td>
<td>Overcast</td>
<td>Mild</td>
<td>High</td>
<td>Strong</td>
<td>Yes</td>
</tr>
<tr>
<td>D13</td>
<td>Overcast</td>
<td>Hot</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D14</td>
<td>Rain</td>
<td>Mild</td>
<td>High</td>
<td>Strong</td>
<td>No</td>
</tr>
</tbody>
</table>
**Example**

- **Learning Phase**

<table>
<thead>
<tr>
<th>Outlook</th>
<th>Play=Yes</th>
<th>Play=No</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sunny</td>
<td>2/9</td>
<td>3/5</td>
</tr>
<tr>
<td>Overcast</td>
<td>4/9</td>
<td>0/5</td>
</tr>
<tr>
<td>Rain</td>
<td>3/9</td>
<td>2/5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Temperature</th>
<th>Play=Yes</th>
<th>Play=No</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hot</td>
<td>2/9</td>
<td>2/5</td>
</tr>
<tr>
<td>Mild</td>
<td>4/9</td>
<td>2/5</td>
</tr>
<tr>
<td>Cool</td>
<td>3/9</td>
<td>1/5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Humidity</th>
<th>Play=Yes</th>
<th>Play=No</th>
</tr>
</thead>
<tbody>
<tr>
<td>High</td>
<td>3/9</td>
<td>4/5</td>
</tr>
<tr>
<td>Normal</td>
<td>6/9</td>
<td>1/5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Wind</th>
<th>Play=Yes</th>
<th>Play=No</th>
</tr>
</thead>
<tbody>
<tr>
<td>Strong</td>
<td>3/9</td>
<td>3/5</td>
</tr>
<tr>
<td>Weak</td>
<td>6/9</td>
<td>2/5</td>
</tr>
</tbody>
</table>

\[ P(\text{Play}=\text{Yes}) = 9/14 \quad P(\text{Play}=\text{No}) = 5/14 \]
Example

- Test Phase

  - Given a new instance, predict its label
    \( x' = (\text{Outlook}=\text{Sunny}, \text{Temperature}=\text{Cool}, \text{Humidity}=\text{High}, \text{Wind}=\text{Strong}) \)

  - Look up tables achieved in the learning phrase

    \[
    \begin{align*}
    P(\text{Outlook}=\text{Sunny} | \text{Play}=\text{Yes}) &= 2/9 & P(\text{Outlook}=\text{Sunny} | \text{Play}=\text{No}) &= 3/5 \\
    P(\text{Temperature}=\text{Cool} | \text{Play}=\text{Yes}) &= 3/9 & P(\text{Temperature}=\text{Cool} | \text{Play}=\text{No}) &= 1/5 \\
    P(\text{Humidity}=\text{High} | \text{Play}=\text{Yes}) &= 3/9 & P(\text{Humidity}=\text{High} | \text{Play}=\text{No}) &= 4/5 \\
    P(\text{Wind}=\text{Strong} | \text{Play}=\text{Yes}) &= 3/9 & P(\text{Wind}=\text{Strong} | \text{Play}=\text{No}) &= 3/5 \\
    P(\text{Play}=\text{Yes}) &= 9/14 & P(\text{Play}=\text{No}) &= 5/14 \\
    \end{align*}
    \]

  - Decision making with the MAP rule

    \[
    \begin{align*}
    P(\text{Yes} | x') &\approx [P(\text{Sunny} | \text{Yes}) P(\text{Cool} | \text{Yes}) P(\text{High} | \text{Yes}) P(\text{Strong} | \text{Yes})] P(\text{Play}=\text{Yes}) = 0.0053 \\
    P(\text{No} | x') &\approx [P(\text{Sunny} | \text{No}) P(\text{Cool} | \text{No}) P(\text{High} | \text{No}) P(\text{Strong} | \text{No})] P(\text{Play}=\text{No}) = 0.0206 \\
    \end{align*}
    \]

    Given the fact \( P(\text{Yes} | x') < P(\text{No} | x') \), we label \( x' \) to be “No”.
Zero conditional probability

- If no example contains the feature value
  - In this circumstance, we face a zero conditional probability problem during test
    \[ \hat{P}(x_1 | c_i) \cdots \hat{P}(a_{jk} | c_i) \cdots \hat{P}(x_n | c_i) = 0 \quad \text{for} \quad x_j = a_{jk}, \quad \hat{P}(a_{jk} | c_i) = 0 \]
  - For a remedy, class conditional probabilities re-estimated with

\[
\hat{P}(a_{jk} | c_i) = \frac{n_c + mp}{n + m} \quad \text{(m-estimate)}
\]

- \( n_c \): number of training examples for which \( x_j = a_{jk} \) and \( c = c_i \)
- \( n \): number of training examples for which \( c = c_i \)
- \( p \): prior estimate (usually, \( p = 1/t \) for \( t \) possible values of \( x_j \))
- \( m \): weight to prior (number of "virtual" examples, \( m \geq 1 \))
Zero conditional probability

Example: \( P(\text{outlook}=\text{overcast}|\text{no})=0 \) in the play-tennis dataset

Adding \( m \) “virtual” examples (\( m: \) up to 1\% of \#training example)

- In this dataset, \# of training examples for the “no” class is 5.
- We can only add \( m=1 \) “virtual” example in our m-estimate remedy.
  - The “outlook” feature can takes only 3 values. So \( p=1/3 \).
  - Re-estimate \( P(\text{outlook}|\text{no}) \) with the m-estimate

\[
P(\text{overcast}|\text{no}) = \frac{0+1*\left(\frac{1}{3}\right)}{5+1} = \frac{1}{18}
\]

\[
P(\text{sunny}|\text{no}) = \frac{3+1*\left(\frac{1}{3}\right)}{5+1} = \frac{5}{9}
\]

\[
P(\text{rain}|\text{no}) = \frac{2+1*\left(\frac{1}{3}\right)}{5+1} = \frac{7}{18}
\]
Numerical Stability

• Recall: NB classifier:

\[ \alpha \prod_{i=1}^{m} P(X_i|Y)P(Y) \]

– Multiplying probabilities can get us into problems!
– Imagine computing the probability of 2000 independent coin flips
– Most programming environments: \((0.5)^{2000}=0\)
Numerical Stability

• Our problem: **Underflow Prevention**
• Recall: \( \log(xy) = \log(x) + \log(y) \)
• better to sum logs of probabilities rather than multiplying probabilities.
• Class with highest final un-normalized log probability score is still the most probable.

\[
c_{NB} = \arg\max_{c_j \in C} \log P(c_j) + \sum_{i \in \text{positions}} \log P(x_i | c_j)
\]
Naïve Bayes: Dealing with Continuous-valued Features

When facing a continuous-valued feature

Conditional probability often modeled with the normal distribution

\[
\hat{P}(x_j | c_i) = \frac{1}{\sqrt{2\pi} \sigma_{ji}} \exp \left( -\frac{(x_j - \mu_{ji})^2}{2\sigma_{ji}^2} \right)
\]

\(\mu_{ji}\): mean (average) of feature values \(x_j\) of examples for which \(c = c_i\)

\(\sigma_{ji}\): standard deviation of feature values \(x_j\) of examples for which \(c = c_i\)

- **Learning Phase:** for \(X = (X_1, \cdots, X_n), \ C = c_1, \cdots, c_L\)

  Output: \(n \times L\) normal distributions and \(P(C = c_i) \ i = 1, \cdots, L\)

- **Test Phase:** Given an unknown instance \(X' = (a'_1, \cdots, a'_n)\)

  - Instead of looking-up tables, calculate conditional probabilities with all the normal distributions achieved in the learning phrase
  - Apply the MAP rule to assign a label (the same as the discrete case)
Naïve Bayes

- **Example: Continuous-valued Features**
  - Temperature is naturally of continuous value.
  
  **Yes:** 25.2, 19.3, 18.5, 21.7, 20.1, 24.3, 22.8, 23.1, 19.8
  
  **No:** 27.3, 30.1, 17.4, 29.5, 15.1
  
  - Estimate mean and variance for each class
  
  $\mu = \frac{1}{N} \sum_{n=1}^{N} x_n, \quad \sigma^2 = \frac{1}{N} \sum_{n=1}^{N} (x_n - \mu)^2$

  $\mu_{Yes} = 21.64, \quad \sigma_{Yes} = 2.35$

  $\mu_{No} = 23.88, \quad \sigma_{No} = 7.09$

  - **Learning Phase:** output two Gaussian models for $P(\text{temp}|C)$

  $\hat{P}(x|\text{Yes}) = \frac{1}{2.35\sqrt{2\pi}} \exp\left(-\frac{(x - 21.64)^2}{2 \times 2.35^2}\right) = \frac{1}{2.35\sqrt{2\pi}} \exp\left(-\frac{(x - 21.64)^2}{11.09}\right)$

  $\hat{P}(x|\text{No}) = \frac{1}{7.09\sqrt{2\pi}} \exp\left(-\frac{(x - 23.88)^2}{2 \times 7.09^2}\right) = \frac{1}{7.09\sqrt{2\pi}} \exp\left(-\frac{(x - 23.88)^2}{50.25}\right)$
Summary

• Naïve Bayes: the conditional independence assumption
  – Training and test are very efficient
  – Two different data types lead to two different learning algorithms
  – Working well sometimes for data violating the assumption!

• A popular generative model
  – Performance competitive to most of state-of-the-art classifiers even in presence of violating independence assumption
  – Many successful applications, e.g., spam mail filtering
  – A good candidate of a base learner in ensemble learning
  – Apart from classification, naïve Bayes can do more...