Tutorial 3

CPSC 340: Machine Learning and Data Mining

Fall 2017
Overview

1 Naive Bayes Classifier

2 Non-Parametric Models
   - Definitions
   - KNN

3 Training, Testing, and Validation Set
Naive Bayes Classifier

- **Naive Bayes** is a probabilistic classifier.
  - Based on Bayes’ theorem.
  - **Strong independence assumption** between features.
Naive Bayes Classifier

- **Naive Bayes** is a probabilistic classifier.
  - Based on Bayes’ theorem.
  - Strong independence assumption between features.

- In the rest of this tutorial,
  - We use \( y_i \) for the label of object \( i \) (element \( i \) of \( y \)).
  - We use \( x_i \) for the features of object \( i \) (row \( i \) of \( X \)).
  - We use \( x_{ij} \) for feature \( j \) of object \( i \).
  - We use \( d \) for the number of features in object \( i \).
Bayes’ rule

\[
p(y_i|x_i) = \frac{p(x_i|y_i)p(y_i)}{p(x_i)}
\]

we want to compare \(P(y=\text{clx}_i)\) for different values of \(c\) and choose the maximum value
Bayes’ rule

\[ p(y_i|x_i) = \frac{p(x_i|y_i)p(y_i)}{p(x_i)} \]

Since the denominator does not depend on \( y_i \), we are only interested in the numerator:

\[ p(y_i|x_i) \propto p(x_i|y_i)p(y_i) \]
Naive Bayes Classifier

- The numerator is equal to the joint probability:

\[ p(x_i | y_i) p(y_i) = p(x_i, y_i) = p(x_{i1}, \ldots, x_{id}, y_i) \]

\[ P(x_{i1}, x_{i2}, \ldots, y_i) = P(x_{i1} | x_{i2}, x_{i3}, \ldots, y_i) \cdot P(x_{i2}, x_{i3}, \ldots, y_i) \]
The numerator is equal to the joint probability:

\[ p(x_i | y_i)p(y_i) = p(x_i, y_i) = p(x_{i1}, \ldots, x_{id}, y_i) \]

Chain rule:

\[
p(x_{i1}, \ldots, x_{id}, y_i) = p(x_{i1} | x_{i2}, \ldots, x_{id}, y_i)p(x_{i2}, \ldots, x_{id}, y_i) \\
= \ldots \\
= p(x_{i1} | x_{i2}, \ldots, x_{id}, y_i)p(x_{i2} | x_{i3}, \ldots, x_{id}, y_i) \ldots p(x_{id} | y_i)p(y_i) \\
\]

\[ P(x_{i1} | y_i) \quad P(x_{i2} | y_i) \quad P(x_{id} | y_i) \quad P(y_i) \]

These are our parameters
Naive Bayes Classifier

- The numerator is equal to the joint probability:

\[ p(x_i | y_i)p(y_i) = p(x_i, y_i) = p(x_1, \ldots, x_id, y_i) \]

- Chain rule:

\[
\begin{align*}
  p(x_1, \ldots, x_id, y_i) &= p(x_1 | x_1, \ldots, x_id, y_i)p(x_1, \ldots, x_id, y_i) \\
  &= \ldots \\
  &= p(x_1 | x_2, \ldots, x_id, y_i)p(x_2 | x_3, \ldots, x_id, y_i) \ldots p(x_id | y_i)p(y_i)
\end{align*}
\]

- Each feature in \( x_i \) is independent of the others given \( y_i \):

\[
p(x_{ij} | x_{ij+1}, \ldots, x_id, y_i) = p(x_{ij} | y_i)
\]
The numerator is equal to the joint probability:

\[ p(x_i | y_i) p(y_i) = p(x_i, y_i) = p(x_{i1}, \ldots, x_{id}, y_i) \]

Chain rule:

\[ p(x_{i1}, \ldots, x_{id}, y_i) = p(x_{i1} | x_{i2}, \ldots, x_{id}, y_i) p(x_{i2}, \ldots, x_{id}, y_i) \]
\[ = \ldots \]
\[ = p(x_{i1} | x_{i2}, \ldots, x_{id}, y_i) p(x_{i2} | x_{i3}, \ldots, x_{id}, y_i) \ldots p(x_{id} | y_i) p(y_i) \]

Each feature in \( x_i \) is independent of the others given \( y_i \):

\[ p(x_{ij} | x_{ij+1}, \ldots, x_{id}, y_i) = p(x_{ij} | y_i) \]

Therefore:

our score for a given \( y_i \)

\[ p(y_i, x_i) \propto p(y_i) \prod_{j=1}^{d} p(x_{ij} | y_i) \]
Problem: Naive Bayes Classifier

<table>
<thead>
<tr>
<th>headache</th>
<th>runny nose</th>
<th>fever</th>
<th>flu</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
</tr>
</tbody>
</table>
We first need to compute our parameters

Prior: $P(\text{flu}=N)$
$$= \frac{3}{6} = \frac{1}{2}$$

Conditional: $P(\text{head}=Y|\text{flu}=N)$
$$= \frac{1}{3}$$
We need

<table>
<thead>
<tr>
<th>Event</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>headache = Y</td>
<td>flu = N</td>
</tr>
<tr>
<td>headache = Y</td>
<td>flu = Y</td>
</tr>
<tr>
<td>runny nose = N</td>
<td>flu = N</td>
</tr>
<tr>
<td>runny nose = N</td>
<td>flu = Y</td>
</tr>
<tr>
<td>fever = Y</td>
<td>flu = N</td>
</tr>
<tr>
<td>fever = Y</td>
<td>flu = Y</td>
</tr>
<tr>
<td>flu = N</td>
<td></td>
</tr>
<tr>
<td>flu = Y</td>
<td></td>
</tr>
</tbody>
</table>
Solution: Naive Bayes Classifier

- We need

<table>
<thead>
<tr>
<th>Event</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p(\text{headache}=Y</td>
<td>\text{flu}=N)$</td>
</tr>
<tr>
<td>$p(\text{headache}=Y</td>
<td>\text{flu}=Y)$</td>
</tr>
<tr>
<td>$p(\text{runny nose}=N</td>
<td>\text{flu}=N)$</td>
</tr>
<tr>
<td>$p(\text{runny nose}=N</td>
<td>\text{flu}=Y)$</td>
</tr>
<tr>
<td>$p(\text{fever}=Y</td>
<td>\text{flu}=N)$</td>
</tr>
<tr>
<td>$p(\text{fever}=Y</td>
<td>\text{flu}=Y)$</td>
</tr>
<tr>
<td>$p(\text{flu}=N)$</td>
<td>$\frac{1}{2}$</td>
</tr>
<tr>
<td>$p(\text{flu}=Y)$</td>
<td>$\frac{1}{2}$</td>
</tr>
</tbody>
</table>

- $p(\text{flu} = N|\text{headache} = Y, \text{runny nose} = N, \text{fever} = Y) \propto p(\text{headache} = Y|\text{flu} = N)p(\text{runny nose} = N|\text{flu} = N)p(\text{fever} = Y|\text{flu} = N)p(\text{flu} = N) = \frac{1}{3} \times \frac{2}{3} \times \frac{1}{3} \times \frac{1}{2} = 0.0370$
Solution: Naive Bayes Classifier

- We need

<table>
<thead>
<tr>
<th>Event</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p(\text{headache}=Y</td>
<td>\text{flu}=N)$</td>
</tr>
<tr>
<td>$p(\text{headache}=Y</td>
<td>\text{flu}=Y)$</td>
</tr>
<tr>
<td>$p(\text{runny nose}=N</td>
<td>\text{flu}=N)$</td>
</tr>
<tr>
<td>$p(\text{runny nose}=N</td>
<td>\text{flu}=Y)$</td>
</tr>
<tr>
<td>$p(\text{fever}=Y</td>
<td>\text{flu}=N)$</td>
</tr>
<tr>
<td>$p(\text{fever}=Y</td>
<td>\text{flu}=Y)$</td>
</tr>
<tr>
<td>$p(\text{flu}=\text{N})$</td>
<td>$\frac{1}{2}$</td>
</tr>
<tr>
<td>$p(\text{flu}=\text{Y})$</td>
<td>$\frac{1}{2}$</td>
</tr>
</tbody>
</table>

- $p(\text{flu}=\text{N}|\text{headache}=\text{Y}, \text{runny nose}=\text{N}, \text{fever}=\text{Y}) \propto \frac{1}{3} \cdot \frac{2}{3} \cdot \frac{1}{3} \cdot \frac{1}{2} = 0.0370$

- $p(\text{flu}=\text{Y}|\text{headache}=\text{Y}, \text{runny nose}=\text{N}, \text{fever}=\text{Y}) \propto \frac{2}{3} \cdot \frac{1}{3} \cdot \frac{2}{3} \cdot \frac{1}{2} = 0.0741$
Solution: Naive Bayes Classifier

- We need

<table>
<thead>
<tr>
<th>Event</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p(\text{headache} = Y</td>
<td>\text{flu} = N)$</td>
</tr>
<tr>
<td>$p(\text{headache} = Y</td>
<td>\text{flu} = Y)$</td>
</tr>
<tr>
<td>$p(\text{runny nose} = N</td>
<td>\text{flu} = N)$</td>
</tr>
<tr>
<td>$p(\text{runny nose} = N</td>
<td>\text{flu} = Y)$</td>
</tr>
<tr>
<td>$p(\text{fever} = Y</td>
<td>\text{flu} = N)$</td>
</tr>
<tr>
<td>$p(\text{fever} = Y</td>
<td>\text{flu} = Y)$</td>
</tr>
<tr>
<td>$p(\text{flu} = N)$</td>
<td>1/2</td>
</tr>
<tr>
<td>$p(\text{flu} = Y)$</td>
<td>1/2</td>
</tr>
</tbody>
</table>

- $p(\text{flu} = N | \text{headache} = Y, \text{runny nose} = N, \text{fever} = Y) \propto p(\text{headache} = Y | \text{flu} = N) p(\text{runny nose} = N | \text{flu} = N) p(\text{fever} = Y | \text{flu} = N) p(\text{flu} = N) = \frac{1}{3} \times \frac{2}{3} \times \frac{1}{3} \times \frac{1}{2} = 0.0370$

- $p(\text{flu} = Y | \text{headache} = Y, \text{runny nose} = N, \text{fever} = Y) \propto p(\text{headache} = Y | \text{flu} = Y) p(\text{runny nose} = N | \text{flu} = Y) p(\text{fever} = Y | \text{flu} = Y) p(\text{flu} = Y) = \frac{2}{3} \times \frac{1}{3} \times \frac{2}{3} \times \frac{1}{2} = 0.0741$

<table>
<thead>
<tr>
<th>headache</th>
<th>runny nose</th>
<th>fever</th>
<th>flu</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
</tr>
</tbody>
</table>
Bayes’ Theorem enables us to reverse probabilities:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$
Problem: Prosecutor’s fallacy

- A crime has been committed in a large city and footprints are found at the scene of the crime. The guilty person matches the footprints, $p(F|G) = 1$. Out of the innocent people, 1% match the footprints by chance, $p(F|\sim G) = 0.01$. A person is interviewed at random and his/her footprints are found to match those at the crime scene. Determine the probability that the person is guilty, or explain why this is not possible, $p(G|F) = ?$

  - Let $F$ be the event that the footprints match.
  - Let $G$ be the event that the person is guilty
  - $\sim G$ be the event that the person is innocent.
Solution: Prosecutor’s fallacy

\[
p(G|F) = \frac{p(F|G)p(G)}{p(F)} = \frac{p(F|G)p(G)}{p(F|G)p(G) + p(F|\sim G)p(\sim G)}
\]
Solution: Prosecutor’s fallacy

\[ p(G|F) = \frac{p(F|G)p(G)}{p(F)} = \frac{p(F|G)p(G)}{p(F|G)p(G) + p(F|\sim G)p(\sim G)} \]

- \( p(G) = ? \rightarrow \text{Impossible!} \)
Definitions

- Parametric Models
  - Fixed number of parameters - learned (estimated) from data
  - More data $\Rightarrow$ More accurate models.
Definitions

- **Parametric Models**
  - Fixed number of parameters - learned (estimated) from data
  - More data ⇒ More accurate models.

- **Non-parametric Models**
  - Number of parameters grows with the amount of data
  - More data  More complex models.
Definitions

- **Parametric Models**
  - Fixed number of parameters - learned (estimated) from data
  - More data $\Rightarrow$ More accurate models.

- **Non-parametric Models**
  - Number of parameters grows with the amount of data
  - More data $\Rightarrow$ More complex models.

- **Parametric or Non-parametric? What are the parameters?**
  - Decision Trees  $P$ (if depth is given)
  - Naive Bayes  $P$ (if features are discrete)
  - KNN  Non-$p$
  - Random Forests  (the number of trees are fixed, but the depth usually varies with data) Non-$p$
  - K-Means Clustering  $P$ ($k$ is given)
k-Nearest Neighbour

- How does it work?
k-Nearest Neighbour

- How does it work?
- What is the effect of k with respect to the fundamental tradeoff in machine learning?
k-Nearest Neighbour

- How does it work?
- What is the effect of $k$ with respect to the fundamental tradeoff in machine learning?
- What is the runtime?
Given training data, we would like to learn a model to minimize error on the testing data.

How do we decide decision tree depth?

We care about test error.

But we can’t look at test data.

So what do we do?????

One answer: Use part of your train data to approximate test error.

Split training objects into training set and validation set:

- Train model on the training data.
- Test model on the validation data.
Cross-Validation

- Isn’t it wasteful to only use part of your data?
- **k-fold cross-validation:**
  - Train on k-1 folds of the data, validate on the other fold.
  - Repeat this k times with different splits, and average the score.

![Diagram of k-fold cross-validation]

**Figure 1:** Adapted from Wikipedia.

- Note: if examples are ordered, split should be random.