Classification: Naïve Bayes Classifier Evaluation

Toon Calders (t.calders@tue.nl)

Sheets are based on the those provided by Tan, Steinbach, and Kumar. *Introduction to Data Mining*
Last Lecture …

• Classification
  • Learning a *model* from *labeled training data* that allows for *predicting the class of unseen test examples*.

• Decision Trees as a model type
  • *Hunt’s algorithm* for induction

• Nearest neighbors classifiers
  • Training set in the model
  • *Distance measure* is central
Outline for Today

• Naïve Bayes Classifier
  • Bayes theorem
  • Class Independence
  • Deduction
• Evaluating Models and Classifiers
  • Accuracy
    – Class imbalance problem
  • Precision/recall
• Issues in classifier learning
Outline for Today

- Naïve Bayes Classifier
  - Bayes theorem
  - Class Independence
  - Deduction
- Evaluating Models and Classifiers
  - Accuracy
    - Class imbalance problem
  - Precision/recall
- Issues in classifier learning
Maximum Likelihood Classifiers

- Based on probability theory: examples in different classes follow different distributions

![Histogram showing different distributions of sepal length]
Maximum Likelihood Classifiers

• “Learn” these different distributions, one for each class.

Class Iris-setosa: Prior probability = 0.33
sepal length: Normal Distribution. Mean = 4.9913 StandardDev = 0.355
sepal width: Normal Distribution. Mean = 3.4015 StandardDev = 0.3925
petal length: Normal Distribution. Mean = 1.4694 StandardDev = 0.1782
petal width: Normal Distribution. Mean = 0.2743 StandardDev = 0.1096

Class Iris-versicolor: Prior probability = 0.33
sepal length: Normal Distribution. Mean = 5.9379 StandardDev = 0.5042
sepal width: Normal Distribution. Mean = 2.7687 StandardDev = 0.3038
petal length: Normal Distribution. Mean = 4.2452 StandardDev = 0.4712
petal width: Normal Distribution. Mean = 1.3097 StandardDev = 0.1915

...
Maximum Likelihood Classifiers

• When a new example arrives
  • For each class, measure its probability,
  • Use Bayes’ theorem to find most likely class that generated it.

Simplified example:

\[ P(X|C_1) = 5\% \quad P(C_1) = 50\% \]
\[ P(X|C_2) = 65\% \quad P(C_2) = 50\% \]

Predict class C2
Recall Bayes’ Theorem

- A probabilistic framework for solving classification problems
- Conditional Probability:

\[ P(A \mid C) = \frac{P(A, C)}{P(C)} \]

- Bayes theorem:

\[ P(C \mid A) = \frac{P(A \mid C)P(C)}{P(A)} \]
### Example of 1-Attribute Bayesian Classifier

<table>
<thead>
<tr>
<th></th>
<th>meningitis</th>
<th>no meningitis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stiff neck</td>
<td>90%</td>
<td>5%</td>
</tr>
<tr>
<td>No stiff neck</td>
<td>10%</td>
<td>95%</td>
</tr>
</tbody>
</table>

\[
P(M) = \frac{1}{100,000} \quad \quad P(\text{no } M) = 1 - \frac{1}{100,000}
\]

- If a patient has stiff neck, what’s the probability he/she has meningitis?

\[
P(M \mid S) = \frac{P(S \mid M)P(M)}{P(S)} = \frac{0.9}{0.9 + 0.05 \times 99999} = 0.00018
\]
Bayesian Classifiers: More Variables

• Consider each attribute and class label as random variables

• Given a record with attributes \((A_1, A_2, \ldots, A_n)\)
  • Goal is to predict class \(C\)
  • Specifically, we want to find the value of \(C\) that maximizes \(P(C| A_1, A_2, \ldots, A_n)\)

• Can we estimate \(P(C| A_1, A_2, \ldots, A_n)\) directly from data?
Bayesian Classifiers

- **Approach:**
  - compute the posterior probability $P(C \mid A_1, A_2, \ldots, A_n)$ for all values of $C$ using the Bayes’ theorem
  
  $$P(C \mid A_1 A_2 \ldots A_n) = \frac{P(A_1 A_2 \ldots A_n \mid C) P(C)}{P(A_1 A_2 \ldots A_n)}$$

  - Choose value of $C$ that maximizes $P(C \mid A_1, A_2, \ldots, A_n)$

  - Equivalent to choosing value of $C$ that maximizes $P(A_1, A_2, \ldots, A_n \mid C) P(C)$

- How to estimate $P(A_1, A_2, \ldots, A_n \mid C)$?
Bayesian Classifiers

• How to estimate \( P(A_1, A_2, \ldots, A_n \mid C) \)?
  - Select all tuples of class \( C \) in training set
  - Count all possible combinations of \( A_1, A_2, \ldots, A_n \)

• However:
  - Not all combinations are present

• Hence:
  - Additional assumptions on the distribution
    - Class independence
Naïve Bayes Classifier

• **Assume independence** among attributes $A_i$ **when class is given**:

  $P(A_1, A_2, \ldots, A_n | C_j) = P(A_1 | C_j) P(A_2 | C_j) \ldots P(A_n | C_j)$

\[
P(C | A_1 A_2 \ldots A_n) = \frac{P(A_1 A_2 \ldots A_n | C) P(C)}{P(A_1 A_2 \ldots A_n)}
\]

\[
= \left( \prod_{i=1}^{n} P(A_i | C) \right) P(C)
\]
Naïve Bayes Classifier

• Learning the model
  • For every class $C$:
    – Estimate the prior $P(C)$
    – For every attribute $A$, for every attribute value $v$ of $A$:
      estimate $P(A=v \mid C)$

• Applying the model
  • Given an example $(v_1, v_2, \ldots, v_n)$
  • Pick the class $C$ that maximizes

$$\left( \prod_{i=1}^{n} P(A_i = v_i \mid C) \right) P(C)$$
Example (DM = class)

<table>
<thead>
<tr>
<th>DB</th>
<th>Stat</th>
<th>DM</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>A</td>
<td>+</td>
</tr>
<tr>
<td>B</td>
<td>B</td>
<td>+</td>
</tr>
<tr>
<td>C</td>
<td>C</td>
<td>-</td>
</tr>
<tr>
<td>A</td>
<td>B</td>
<td>-</td>
</tr>
<tr>
<td>A</td>
<td>B</td>
<td>+</td>
</tr>
</tbody>
</table>

• Model:
Example (DM = class)

<table>
<thead>
<tr>
<th>DB</th>
<th>Stat</th>
<th>DM</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>A</td>
<td>+</td>
</tr>
<tr>
<td>B</td>
<td>B</td>
<td>+</td>
</tr>
<tr>
<td>C</td>
<td>C</td>
<td>-</td>
</tr>
<tr>
<td>A</td>
<td>B</td>
<td>-</td>
</tr>
<tr>
<td>A</td>
<td>B</td>
<td>+</td>
</tr>
</tbody>
</table>

- **Model:**

  **Class = + (60%)**
  - **DB:**
    - A: 1
    - B: 2
    - C: 0
  - **Stat:**
    - A: 1
    - B: 2
    - C: 0

  **Class = - (40%)**
  - **DB:**
    - A: 1
    - B: 0
    - C: 1
  - **Stat:**
    - A: 0
    - B: 1
    - C: 1
Example (DM = class)

<table>
<thead>
<tr>
<th>DB</th>
<th>Stat</th>
<th>DM</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>A</td>
<td>+</td>
</tr>
<tr>
<td>B</td>
<td>B</td>
<td>+</td>
</tr>
<tr>
<td>C</td>
<td>C</td>
<td>-</td>
</tr>
<tr>
<td>A</td>
<td>B</td>
<td>-</td>
</tr>
<tr>
<td>A</td>
<td>B</td>
<td>+</td>
</tr>
</tbody>
</table>

- Model:
  - **Class = + (60%)**
    - **DB:**
      - A: 1
      - B: 2
      - C: 0
    - **Stat:**
      - A: 1
      - B: 2
      - C: 0
  - **Class = - (40%)**
    - **DB:**
      - A: 1
      - B: 0
      - C: 1
    - **Stat:**
      - A: 0
      - B: 1
      - C: 1

- Prediction for (DB=A, Stat=B)?
Example (DM = class)

- Model:
  
  **Class = + (60%)**
  
  DB: A 1
  B 2
  C 0
  
  Stat: A 1
  B 2
  C 0
  
  **Class = - (40%)**
  
  DB: A 1
  B 0
  C 1
  
  Stat: A 0
  B 1
  C 1

- Prediction for (DB=A, Stat=B)?

  Class - : 40% x 50% x 50% = 0.1
  
  Class + : 60% x 33% x 66% = 0.13...
Continuous attributes?

- For continuous attributes:
  - Discretize into bins

- Split: \((A < v)\) or \((A > v)\)

- Probability density estimation:
  - Assume attribute follows a normal distribution
  - Use data to estimate parameters of distribution (e.g., mean and standard deviation)
  - Once probability distribution is known, can use it to estimate the conditional probability \(P(A_i|c)\)
Estimating the Parameters

- Density function expresses relative probability of a point.
- For Normally distributed data with mean $\mu$ and standard deviation $\sigma$, the density function is:

$$
\varphi_{\mu,\sigma}(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}
$$

- $\mu$ and $\sigma$ can be estimated from data, for each class $C$ separately
- Use $\varphi_{\mu,\sigma}(x)$ for $P(X \mid C)$
Example: Mix of Attributes

<table>
<thead>
<tr>
<th>Tid</th>
<th>Refund</th>
<th>Marital Status</th>
<th>Taxable Income</th>
<th>Evade</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Yes</td>
<td>Single</td>
<td>125K</td>
<td>No</td>
</tr>
<tr>
<td>2</td>
<td>No</td>
<td>Married</td>
<td>100K</td>
<td>No</td>
</tr>
<tr>
<td>3</td>
<td>No</td>
<td>Single</td>
<td>70K</td>
<td>No</td>
</tr>
<tr>
<td>4</td>
<td>Yes</td>
<td>Married</td>
<td>120K</td>
<td>No</td>
</tr>
<tr>
<td>5</td>
<td>No</td>
<td>Divorced</td>
<td>95K</td>
<td>Yes</td>
</tr>
<tr>
<td>6</td>
<td>No</td>
<td>Married</td>
<td>60K</td>
<td>No</td>
</tr>
<tr>
<td>7</td>
<td>Yes</td>
<td>Divorced</td>
<td>220K</td>
<td>No</td>
</tr>
<tr>
<td>8</td>
<td>No</td>
<td>Single</td>
<td>85K</td>
<td>Yes</td>
</tr>
<tr>
<td>9</td>
<td>No</td>
<td>Married</td>
<td>75K</td>
<td>No</td>
</tr>
<tr>
<td>10</td>
<td>No</td>
<td>Single</td>
<td>90K</td>
<td>Yes</td>
</tr>
</tbody>
</table>

naive Bayes Classifier:

P(Refund=Yes|No) = 3/7
P(Refund=No|No) = 4/7
P(Refund=Yes|Yes) = 0
P(Refund=No|Yes) = 1
P(Marital Status=Single|No) = 2/7
P(Marital Status=Divorced|No)=1/7
P(Marital Status=Married|No) = 4/7
P(Marital Status=Single|Yes) = 2/7
P(Marital Status=Divorced|Yes)=1/7
P(Marital Status=Married|Yes) = 0

For taxable income:
If class=No: sample mean=110
sample variance=2975
If class=Yes: sample mean=90
sample variance=25
Example: Mix of Attributes

Given test record:

\[ X = (\text{Refund} = \text{No}, \text{Divorced}, \text{Income} = 120K) \]

naive Bayes Classifier:

\[
\begin{align*}
P(\text{Refund}=\text{Yes}|\text{No}) &= \frac{3}{7} \\
P(\text{Refund}=\text{No}|\text{No}) &= \frac{4}{7} \\
P(\text{Refund}=\text{Yes}|\text{Yes}) &= 0 \\
P(\text{Refund}=\text{No}|\text{Yes}) &= 1 \\
P(\text{Marital Status}=\text{Single}|\text{No}) &= \frac{2}{7} \\
P(\text{Marital Status}=\text{Divorced}|\text{No}) &= \frac{1}{7} \\
P(\text{Marital Status}=\text{Married}|\text{No}) &= \frac{4}{7} \\
P(\text{Marital Status}=\text{Single}|\text{Yes}) &= \frac{2}{7} \\
P(\text{Marital Status}=\text{Divorced}|\text{Yes}) &= \frac{1}{7} \\
P(\text{Marital Status}=\text{Married}|\text{Yes}) &= 0
\end{align*}
\]

For taxable income:

<table>
<thead>
<tr>
<th>Class</th>
<th>Sample Mean</th>
<th>Sample Variance</th>
</tr>
</thead>
<tbody>
<tr>
<td>No</td>
<td>110</td>
<td>2975</td>
</tr>
<tr>
<td>Yes</td>
<td>90</td>
<td>25</td>
</tr>
</tbody>
</table>

\[
P(X|\text{No})P(\text{No}) > P(X|\text{Yes})P(\text{Yes})
\]

Hence, \( P(\text{No}|X) > P(\text{Yes}|X) \)

Prediction: Class = No
Naïve Bayes Classifier

• If one of the conditional probability is zero, then the entire expression becomes zero

• Probability estimation:

Original: \( P(A_i \mid C) = \frac{N_{ic}}{N_c} \)

Laplace: \( P(A_i \mid C) = \frac{N_{ic} + 1}{N_c + c} \)

m-estimate: \( P(A_i \mid C) = \frac{N_{ic} + mp}{N_c + m} \)

c: number of classes
p: prior probability
m: parameter
Naïve Bayes (Summary)

- Robust to isolated noise points
- Handle missing values by ignoring the instance during probability estimate calculations
- Robust to irrelevant attributes
- Independence assumption may not hold for some attributes
  - Use other techniques such as Bayesian Belief Networks (BBN)
Outline

• Naïve Bayes Classifier
  • Bayes theorem
  • Class Independence
  • Deduction

• Evaluating Models and Classifiers
  • Accuracy
    – Class imbalance problem
  • Precision/recall

• Issues in classifier learning
Metrics for Performance Evaluation

- Focus on the predictive capability of a model
  - Rather than how fast it takes to classify or build models, scalability, etc.
- **Confusion Matrix:**

<table>
<thead>
<tr>
<th>ACTUAL CLASS</th>
<th>PREDICTED CLASS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Class=Yes</td>
</tr>
<tr>
<td>Class=Yes</td>
<td>a</td>
</tr>
<tr>
<td>Class=No</td>
<td>c</td>
</tr>
</tbody>
</table>

- **a:** TP (true positive)
- **b:** FN (false negative)
- **c:** FP (false positive)
- **d:** TN (true negative)
### Metrics for Performance Evaluation...

<table>
<thead>
<tr>
<th>ACTUAL CLASS</th>
<th>PREDICTED CLASS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Class=Yes</td>
</tr>
<tr>
<td>Class=Yes</td>
<td>a (TP)</td>
</tr>
<tr>
<td>Class=No</td>
<td>c (FP)</td>
</tr>
</tbody>
</table>

- **Most widely-used metric:**

\[
\text{Accuracy} = \frac{a + d}{a + b + c + d} = \frac{TP + TN}{TP + TN + FP + FN}
\]
Limitation of Accuracy

- Consider a 2-class problem
  - Number of Class 0 examples = 9990
  - Number of Class 1 examples = 10

- If model predicts everything to be class 0, accuracy is 9990/10000 = 99.9%
  - Accuracy is misleading because model does not detect any class 1 example
### Cost Matrix

| ACTUAL CLASS | PREDICTED CLASS | C(i|j) | Class=Yes | Class=No |
|--------------|----------------|-------|-----------|----------|
| Class=Yes    | C(Yes|Yes)       | C(No|Yes) |
| Class=No     | C(Yes|No)        | C(No|No)  |

C(i|j): Cost of misclassifying class j example as class i
### Computing Cost of Classification

#### Cost Matrix

<table>
<thead>
<tr>
<th>ACTUAL CLASS</th>
<th>PREDICTED CLASS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>+</td>
</tr>
<tr>
<td>+</td>
<td>-1</td>
</tr>
<tr>
<td>-</td>
<td>1</td>
</tr>
</tbody>
</table>

#### Model $M_1$

<table>
<thead>
<tr>
<th>ACTUAL CLASS</th>
<th>PREDICTED CLASS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>+</td>
</tr>
<tr>
<td>+</td>
<td>150</td>
</tr>
<tr>
<td>-</td>
<td>60</td>
</tr>
</tbody>
</table>

Accuracy = 80%
Cost = 3910

#### Model $M_2$

<table>
<thead>
<tr>
<th>ACTUAL CLASS</th>
<th>PREDICTED CLASS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>+</td>
</tr>
<tr>
<td>+</td>
<td>250</td>
</tr>
<tr>
<td>-</td>
<td>5</td>
</tr>
</tbody>
</table>

Accuracy = 90%
Cost = 4255
### Cost vs Accuracy

<table>
<thead>
<tr>
<th>Count</th>
<th>PREDICTED CLASS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Class=Yes</td>
</tr>
<tr>
<td>Class=Yes</td>
<td>a</td>
</tr>
<tr>
<td>Class=No</td>
<td>c</td>
</tr>
</tbody>
</table>

\[ N = a + b + c + d \]

Accuracy is proportional to cost if:
1. \( C(\text{Yes} | \text{No}) = C(\text{No} | \text{Yes}) = q \)
2. \( C(\text{Yes} | \text{Yes}) = C(\text{No} | \text{No}) = p \)

\[ \text{Accuracy} = \frac{(a + d)}{N} \]

Cost:
\[
\begin{align*}
\text{Cost} &= p \ (a + d) + q \ (b + c) \\
&= p \ (a + d) + q \ (N - a - d) \\
&= q \ N - (q - p)(a + d) \\
&= N \ [q - (q-p) \times \text{Accuracy}] 
\end{align*}
\]
Cost-Sensitive Measures

Precision (p) = \frac{a}{a + c}

Recall (r) = \frac{a}{a + b}

F - measure (F) = \frac{2rp}{r + p} = \frac{2a}{2a + b + c}

- Precision is biased towards \text{C(Yes|Yes)} & \text{C(Yes|No)}
- Recall is biased towards \text{C(Yes|Yes)} & \text{C(No|Yes)}
- F-measure is biased towards all except \text{C(No|No)}

Weighted Accuracy = \frac{w_1 a + w_4 d}{w_1 a + w_2 b + w_3 c + w_4 d}
Performance of Classifiers

• Accuracy, Precision, Recall:
  • Suitable for *model* selection

• How to compare *classifiers* on a dataset?
  • E.g., Naïve Bayes vs Decision tree

• Make claims like
  • “On dataset D, a Naïve Bayes Classifier can achieve an accuracy of x%”
Performance of Classifiers

• Holdout
  • Reserve 2/3 for training and 1/3 for testing

• Random subsampling
  • Repeated holdout

• Cross validation
  • Partition data into k disjoint subsets
  • k-fold: train on k-1 partitions, test on the remaining one
  • Leave-one-out: k=n
Outline

• Naïve Bayes Classifier
  • Bayes theorem
  • Class Independence
  • Deduction
• Evaluating Classifiers
  • Accuracy
    – Class imbalance problem
  • Precision/recall, AUC
• Issues in classifier learning
Issues in Classifier Learning

• Class imbalance problem

• Training and testing distribution are different

• Only positive examples have been given

• Concept drift

• …
Exercises

• Decision tree: p. 198, Chapter 4, ex. 2

• Naïve Bayes: p. 318, Chapter 5, ex 7

• All: p. 325 & 326, Chapter 5, ex. 23
Compute Gini-indices

What is the best split?

Why is it a bad idea to split on CustomerID?
Give the model the Naïve Bayesian classifier learns

a) Without m-estimate
b) With m-estimate; p=1/2, m=4
c) Predict in both cases the class of (A=0, B=1, C=0)

Table 5.1. Data set for Exercise 7.

<table>
<thead>
<tr>
<th>Record</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>+</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>−</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>−</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>−</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>+</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>+</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>−</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>−</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>+</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>+</td>
</tr>
</tbody>
</table>
Which classifier is best suited for these data?

Give for every dataset one of:

a) k-nearest neighbor
b) Decision tree
c) Naïve Bayesian classifier