Data Mining I
Classification, Part 1

Heiko Paulheim, Volha Bryl, Robert Meusel, Oliver Lehmberg
Outline

1. What is Classification?
2. k Nearest Neighbors
3. Naïve Bayes
4. Decision Trees
5. Evaluating Classification
6. The Overfitting Problem
7. Rule Learning
8. Other Classification Approaches
9. Parameter Tuning
A Couple of Questions

• What is this?
• Why do you know?
• How have you come to that knowledge?
Introductory Example

- Learning a new concept, e.g., "Tree"

  "tree"
  "tree"
  "tree"
  "tree"
  "not a tree"
  "not a tree"
  "not a tree"
  "not a tree"
Introductory Example

- Example: learning a new concept, e.g., "Tree"
  - we look at (positive and negative) examples
  - ...and derive a model
    - e.g., "Trees are big, green plants"

- Goal: Classification of new instances

"tree?"

Warning: Models are only approximating examples! Not guaranteed to be correct or complete!
What is Classification?

- Classic programming:
  - if more than 10 orders/year and more than $100k spent
    set customer.isPremiumCustomer = true

- The prevalent style of programming computers
  - works well as long as we know the rules
  - e.g.: what makes a customer a premium customer?

![Diagram showing the process of giving instructions and computing results](image)
What is Classification?

- Sometimes, it's not so easy
- E.g., due to missing knowledge
  - if customer is likely to order a new phone
    send advertisement for new phones
- E.g., due to difficult formalization as an algorithm
  - if customer review is angry
    send apology
What is Classification?

• A different paradigm:
  – User provides computer with examples
  – Computer finds model by itself
  – Notion: the computer *learns* from examples (term: *machine learning*)

• Example
  – labeled examples of angry and non-angry customer reviews
  – computer finds model for telling if a customer is angry

[Diagram: Provide examples → computer → build model → compute results]
Classification: Formal Definition

- **Given:**
  - a set of labeled records, consisting of
    - data fields (a.k.a. attributes or features)
    - a class label (e.g., true/false)

- **Generate**
  - a function \( f(r) \)
    - input: a record
    - output: a class label
  - which can be used for classifying previously unseen records

- **Variants:**
  - single class problems (e.g., only true/false)
  - multi class problems
  - multi label problems (more than one class per record, not covered in this lecture)
  - hierarchical multi label problems (with class hierarchy, e.g., product categories)
The Classification Workflow

Training Set

<table>
<thead>
<tr>
<th>Tid</th>
<th>Attrib1</th>
<th>Attrib2</th>
<th>Attrib3</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Yes</td>
<td>Large</td>
<td>125K</td>
<td>No</td>
</tr>
<tr>
<td>2</td>
<td>No</td>
<td>Medium</td>
<td>100K</td>
<td>No</td>
</tr>
<tr>
<td>3</td>
<td>No</td>
<td>Small</td>
<td>70K</td>
<td>No</td>
</tr>
<tr>
<td>4</td>
<td>Yes</td>
<td>Medium</td>
<td>120K</td>
<td>No</td>
</tr>
<tr>
<td>5</td>
<td>No</td>
<td>Large</td>
<td>95K</td>
<td>Yes</td>
</tr>
<tr>
<td>6</td>
<td>No</td>
<td>Medium</td>
<td>60K</td>
<td>No</td>
</tr>
<tr>
<td>7</td>
<td>Yes</td>
<td>Large</td>
<td>220K</td>
<td>No</td>
</tr>
<tr>
<td>8</td>
<td>No</td>
<td>Small</td>
<td>85K</td>
<td>Yes</td>
</tr>
<tr>
<td>9</td>
<td>No</td>
<td>Medium</td>
<td>75K</td>
<td>No</td>
</tr>
<tr>
<td>10</td>
<td>No</td>
<td>Small</td>
<td>90K</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Unseen Records

<table>
<thead>
<tr>
<th>Tid</th>
<th>Attrib1</th>
<th>Attrib2</th>
<th>Attrib3</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>No</td>
<td>Small</td>
<td>55K</td>
<td>?</td>
</tr>
<tr>
<td>12</td>
<td>Yes</td>
<td>Medium</td>
<td>80K</td>
<td>?</td>
</tr>
<tr>
<td>13</td>
<td>Yes</td>
<td>Large</td>
<td>110K</td>
<td>?</td>
</tr>
<tr>
<td>14</td>
<td>No</td>
<td>Small</td>
<td>95K</td>
<td>?</td>
</tr>
<tr>
<td>15</td>
<td>No</td>
<td>Large</td>
<td>67K</td>
<td>?</td>
</tr>
</tbody>
</table>

Learning algorithm

Induction

Learn Model

Apply Model

Deduction
Classification Applications – Examples

• Attributes: a set of symptoms (headache, sore throat...)  
  – class: does the patient suffer from disease X?

• Attributes: the values in your tax declaration  
  – class: are you trying to cheat?

• Attributes: your age, income, debts, …  
  – class: are you getting credit by your bank?

• Attributes: the countries you phoned with in the last 6 months  
  – class: are you a terrorist?
Classification Applications – Examples

• Attributes: words in a product review
  – Class: Is it a fake review?

• Attributes: words and header fields of an e-mail
  – Class: Is it a spam e-mail?
Classification Applications – Examples

• A controversial example
  – Class: whether you are searched by the police
  – Class: whether you are selected at the airport for an extra check

http://lubbockonline.com/stories/030609/loc_405504016.shtml
Classification Algorithms

• Recap:
  – we give the computer a set of labeled examples
  – the computer learns to classify new (unlabeled) examples

• How does that work?
k Nearest Neighbors

• Problem
  – find out what the weather is in a certain place
  – where there is no weather station
  – how could you do that?
k Nearest Neighbors

- Idea: use the average of the nearest stations

- Example:
  - 3x sunny
  - 2x cloudy
  - result: sunny

- Approach is called
  - “k nearest neighbors”
  - where k is the number of neighbors to consider
  - in the example: k=5
  - in the example: “near” denotes geographical proximity
k Nearest Neighbors

• Further examples:
  • Is a customer going to buy a product?
    → have similar customers bought that product?
  • What party are you going to vote for?
    → what party do your (closest) friends/family members vote for?
  • Is a film going to win an oscar?
    → have similar films won an oscar?

• and so on...
Experiment

• Try to predict: do you want to watch “The Hobbit (The Battle of the Five Armies)”?

• Binary attributes: have you watched these 2014 films?
  – Captain America: The Winter Soldier
  – The Grand Budapest Hotel
  – The LEGO Movie
  – Transformers: Age of Extinction
  – Divergent
  – Neighbors
  – Noah
  – Rio 2
Recap: Similarity and Distance

• k Nearest Neighbors
  – requires a notion of similarity (i.e., what is “near”?)

• Review: similarity measures for clustering
  – similarity of individual data values
  – similarity of data points

• Think about scales and normalization!

• Which similarity measure was used in our experiment?
  – we could have used different ones
  – probably with different results
Nearest-Neighbor Classifiers

- Requires three things
  - The set of stored records
  - A distance metric to compute distance between records
  - The value of $k$, the number of nearest neighbors to retrieve
To classify an unknown record:

- **Compute distance** to each training record
- Identify k nearest neighbors
- Use **class labels of nearest neighbors** to determine the class label of unknown record
  - by taking majority vote
  - by weighing the vote according to distance
Definition of the k Nearest Neighbors

The k nearest neighbors of a record $x$ are data points that have the k smallest distance to $x$.

(a) 1-nearest neighbor  
(b) 2-nearest neighbor  
(c) 3-nearest neighbor
Choosing a good Value for k

- If $k$ is too small, sensitive to noise points
- If $k$ is too large, neighborhood may include points from other classes

- Rule of thumb: Test $k$ values between 1 and 10.
Discussion of K-Nearest Neighbor

- Often very accurate
- ... but slow as training data needs to be searched
- Can handle decision boundaries which are not parallel to the axes
- Assumes all attributes are equally important
  - Remedy: Attribute selection or using attribute weights
Decision Boundaries of a k-NN Classifier

- $k=1$
- Single noise points have influence on model
Decision Boundaries of a k-NN Classifier

- $k=3$
- Boundaries become smoother
- Influence of noise points is reduced
KNN in RapidMiner
Applying the Model

[Diagram showing the main process with blocks for Training Data, k-NN, Apply Model, and Unseen Data connected with arrows indicating flow.]
Bayes Classifier

- Based on Bayes Theorem
- Thomas Bayes (1701-1761)
  - British mathematician and priest
  - tried to formally prove the existence of God
- Bayes Theorem
  - important theorem in probability theory
  - was only published after Bayes' death
Conditional Probability and Bayes Theorem

• A probabilistic framework for solving classification problems

• Conditional Probability:

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

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

• Bayes theorem:

\[ P(C|A) = \frac{P(A|C)P(C)}{P(A)} \]
Conditional Probability and Bayes Theorem

• Bayes Theorem
  – Computes one conditional probability $P(C|A)$ out of another $P(A|C)$
  – given that the base probabilities $P(A)$ and $P(C)$ are known

• Useful in situations where $P(C|A)$ is unknown
  – while $P(A|C)$, $P(A)$ and $P(C)$ are known or easy to determine/estimate

• Example:
  – Given a symptom, what's the probability that I have a certain disease?
Example of Bayes Theorem

- Given:
  - A doctor knows that meningitis causes stiff neck 50% of the time
  - Prior probability of any patient having meningitis is 1/50,000
  - Prior probability of any patient having stiff neck is 1/20

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

\[
P(M|S) = \frac{P(S|M)P(M)}{P(S)} = \frac{0.5 \times 1/50000}{1/20} = 0.0002
\]
Example of Bayes' Theorem

http://xkcd.com/1236/
Example of Bayes Theorem

• ELISA Test
  – the most common test for HIV

• Numbers:
  – If you're infected, ELISA shows a positive result with p=99.9%
  – If you're not infected, ELISA shows a negative result with p=99.5%

• Assume you have a positive test
  – What's the probability that you're infected with HIV?

• Make a guess!
Example of Bayes Theorem

• We want to know \( P(\text{HIV}|\text{pos}) \)
  – Bayes theorem:
    \[
P(\text{HIV}|\text{pos}) = \frac{P(\text{pos}|\text{HIV})P(\text{HIV})}{P(\text{pos})}
    \]

• We still need \( P(\text{pos}) \)
  – the probability of a positive test
    \[
P(\text{pos}) = P(\text{pos}|\text{HIV} \lor \neg\text{HIV})
    = P(\text{pos}|\text{HIV}) \cdot P(\text{HIV}) + P(\text{pos}|\neg\text{HIV}) \cdot P(\neg\text{HIV})
    \]

• Putting the pieces together:
  \[
P(\text{HIV}|\text{pos}) = \frac{P(\text{pos}|\text{HIV})P(\text{HIV})}{P(\text{pos}|\text{HIV}) \cdot P(\text{HIV}) + P(\text{pos}|\neg\text{HIV}) \cdot P(\neg\text{HIV})}
  \]

0.1% in Germany
Example of Bayes Theorem

- Now: numbers

\[
P(HIV|pos) = \frac{P(pos|HIV)P(HIV)}{P(pos|HIV)P(HIV)+P(pos|\neg HIV)P(\neg HIV)}
\]

\[
= \frac{0.999 \cdot 0.001}{0.999 \cdot 0.001 + 0.005 \cdot 0.999} = 0.167
\]

- That means:
  - at more than 80% probability, you are still healthy, given a positive test!

- Reason:
  - low overall apriori probability of being HIV positive
Bayesian Classifiers

• 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 the data?
Bayesian Classifiers

• Approach:
  – compute the 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 \( P(A_1, A_2, \ldots, A_n \mid C) \)?
Naïve Bayes Classifier

• Assume independence among attributes $A_i$ when class is given:
  – $P(A_1, A_2, \ldots, A_n | C) = P(A_1 | C_j) P(A_2 | C_j) \cdots P(A_n | C_j)$
  – Can estimate $P(A_i | C_j)$ for all $A_i$ and $C_j$
  – New point is classified to $C_j$ if $P(C_j) \prod P(A_i | C_j)$ is maximal
How to Estimate Probabilities from Data?

<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>

- Class: \( P(C) = \frac{N_c}{N} \)
  - e.g., \( P(\text{No}) = \frac{7}{10} \), \( P(\text{Yes}) = \frac{3}{10} \)

- For discrete attributes:
  \( P(A_i | C_k) = \frac{|A_{ik}|}{N_c} \)
  - where \(|A_{ik}|\) is number of instances having attribute \(A_i\) and belongs to class \(C_k\)

Examples:

\( P(\text{Status=Married}|\text{No}) = \frac{4}{7} \)
\( P(\text{Refund=Yes}|\text{Yes}) = 0 \)
How to Estimate Probabilities from Data?

• For continuous attributes:
  – **Discretize** the range into bins
    • one binary attribute per bin
    • violates independence assumption
  – **Two-way split**: \((A < v)\) or \((A > v)\)
    • choose only one of the two splits as new attribute
  – **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 \(P(A_i|c)\)
### How to Estimate Probabilities from Data?

- **Normal distribution:**
  \[
  P(A_i \mid c_j) = \frac{1}{\sqrt{2\pi\sigma^2_{ij}}} e^{-\frac{(A_i - \mu_j)^2}{2\sigma^2_{ij}}}
  \]
  - One for each \((A_i, c_i)\) pair

- **For (Income, Class=No):**
  - If Class=No
    - sample mean = 110
    - sample variance = 2975

\[
P(\text{Income} = 120 \mid \text{No}) = \frac{1}{\sqrt{2\pi(54.54)}} e^{-\frac{(120-110)^2}{2(2975)}} = 0.0072
\]
How to Estimate Probabilities from Data?

- Example visualization:
  - normal distribution
  - mean = 110
  - variance = 2975

- $P(\text{Income}=120|\text{No}) = 0.0072$
Example of Naïve Bayes Classifier

Given a Test Record:

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

naive Bayes Classifier:

- \[ P(X|\text{Class}=\text{No}) = P(\text{Refund}=\text{No}|\text{Class}=\text{No}) \times P(\text{Married}| \text{Class}=\text{No}) \times P(\text{Income}=120K| \text{Class}=\text{No}) \]
  \[ = \frac{4}{7} \times \frac{4}{7} \times 0.0072 = 0.0024 \]

- \[ P(X|\text{Class}=\text{Yes}) = P(\text{Refund}=\text{No}| \text{Class}=\text{Yes}) \times P(\text{Married}| \text{Class}=\text{Yes}) \times P(\text{Income}=120K| \text{Class}=\text{Yes}) \]
  \[ = 1 \times 0 \times 1.2 \times 10^{-9} = 0 \]

Since \( P(X|\text{No})P(\text{No}) > P(X|\text{Yes})P(\text{Yes}) \)
Therefore \( P(\text{No}|X) > P(\text{Yes}|X) \)
\[ \Rightarrow \text{Class} = \text{No} \]
Handling missing values

- Missing values may occur in training and classification examples.
- **Training:** Instance is not included in frequency count for attribute value-class combination.
- **Classification:** Attribute will be omitted from calculation.
- **Example:**

```plaintext
<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>15</td>
<td>No</td>
<td>?</td>
<td>120k</td>
<td>?</td>
</tr>
</tbody>
</table>
```

Likelihood of “yes” = 1 * (1.2 * 10^{-9}) = 1.2 * 10^{-9}

Likelihood of “no” = 4/7 * 0.0072 = 0.0041
From Likelihoods to Probabilities

• A person can either evade or not
  – so why do the likelihoods not add up to 1?

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

We have ignored the denominator so far!
  – however, it is the same for all classes
  – so we can simply normalize to 1:

Likelihood of “yes” = 1 * (1.2 * 10^{-9}) = 1.2 * 10^{-9}
Likelihood of “no” = 4/7 * 0.0072 = 0.0041
P(“yes”) = 1.2 * 10^{-9} / (1.2 * 10^{-9} + 0.0041) = 0.0000003
P(“no”) = 0.0041 / (1.2 * 10^{-9} + 0.0041) = 0.9999997
Zero Frequency Problem

- If one of the conditional probabilities is zero, then the entire expression becomes zero.
- And it is not unlikely that an exactly same data point has not yet been observed.
- Probability estimation:

\[
\text{Original: } P(A_i|C) = \frac{N_{ic}}{N_c}
\]

\[
\text{Laplace: } P(A_i|C) = \frac{N_{ic} + 1}{N_c + c}
\]

\(c\): number of classes
Naïve Bayes in RapidMiner
### Naïve Bayes in RapidMiner

#### Attribute Distribution Table

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Parameter</th>
<th>no</th>
<th>yes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Outlook</td>
<td>value=rain</td>
<td>0.392</td>
<td>0.331</td>
</tr>
<tr>
<td>Outlook</td>
<td>value=overcast</td>
<td>0.014</td>
<td></td>
</tr>
<tr>
<td>Outlook</td>
<td>value=sunny</td>
<td>0.581</td>
<td></td>
</tr>
<tr>
<td>Outlook</td>
<td>value=unknown</td>
<td>0.014</td>
<td></td>
</tr>
<tr>
<td>Temperature</td>
<td>mean</td>
<td>74.600</td>
<td></td>
</tr>
<tr>
<td>Temperature</td>
<td>standard deviation</td>
<td>7.893</td>
<td></td>
</tr>
<tr>
<td>Humidity</td>
<td>mean</td>
<td>84</td>
<td></td>
</tr>
<tr>
<td>Humidity</td>
<td>standard deviation</td>
<td>9.618</td>
<td></td>
</tr>
<tr>
<td>Wind</td>
<td>value=true</td>
<td>0.589</td>
<td></td>
</tr>
<tr>
<td>Wind</td>
<td>value=false</td>
<td>0.397</td>
<td></td>
</tr>
<tr>
<td>Wind</td>
<td>value=unknown</td>
<td>0.014</td>
<td></td>
</tr>
</tbody>
</table>

#### Attribute Distribution Graph

- **Attribute:** Humidity
- **Distribution:** Normal
- **Density:** Two peaks

---

**09/17/14**

Paulheim, Bryl, Meusel, Lehmborg
Naïve Bayes in RapidMiner
Naïve Bayes in RapidMiner

<table>
<thead>
<tr>
<th>Row No.</th>
<th>Play</th>
<th>confidence(no)</th>
<th>confidence(yes)</th>
<th>prediction(Play)</th>
<th>Outlook</th>
<th>Temperature</th>
<th>Humidity</th>
<th>Wind</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>yes</td>
<td>0.711</td>
<td>0.289</td>
<td>no</td>
<td>sunny</td>
<td>85</td>
<td>85</td>
<td>false</td>
</tr>
<tr>
<td>2</td>
<td>no</td>
<td>0.058</td>
<td>0.942</td>
<td>yes</td>
<td>overcast</td>
<td>80</td>
<td>90</td>
<td>true</td>
</tr>
<tr>
<td>3</td>
<td>yes</td>
<td>0.014</td>
<td>0.986</td>
<td>yes</td>
<td>overcast</td>
<td>83</td>
<td>78</td>
<td>false</td>
</tr>
<tr>
<td>4</td>
<td>yes</td>
<td>0.412</td>
<td>0.588</td>
<td>yes</td>
<td>rain</td>
<td>70</td>
<td>96</td>
<td>false</td>
</tr>
<tr>
<td>5</td>
<td>yes</td>
<td>0.460</td>
<td>0.540</td>
<td>yes</td>
<td>rain</td>
<td>68</td>
<td>80</td>
<td>false</td>
</tr>
<tr>
<td>6</td>
<td>no</td>
<td>0.336</td>
<td>0.664</td>
<td>yes</td>
<td>rain</td>
<td>65</td>
<td>70</td>
<td>true</td>
</tr>
<tr>
<td>7</td>
<td>yes</td>
<td>0.010</td>
<td>0.990</td>
<td>no</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>no</td>
<td>0.596</td>
<td>0.404</td>
<td>no</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>yes</td>
<td>0.248</td>
<td>0.752</td>
<td>yes</td>
<td>sunny</td>
<td>69</td>
<td>70</td>
<td>false</td>
</tr>
<tr>
<td>10</td>
<td>no</td>
<td>0.407</td>
<td>0.593</td>
<td>yes</td>
<td>sunny</td>
<td>75</td>
<td>80</td>
<td>false</td>
</tr>
<tr>
<td>11</td>
<td>yes</td>
<td>0.496</td>
<td>0.504</td>
<td>yes</td>
<td></td>
<td></td>
<td></td>
<td>true</td>
</tr>
<tr>
<td>12</td>
<td>yes</td>
<td>0.038</td>
<td>0.962</td>
<td>yes</td>
<td></td>
<td></td>
<td></td>
<td>true</td>
</tr>
<tr>
<td>13</td>
<td>no</td>
<td>0.027</td>
<td>0.973</td>
<td>yes</td>
<td>overcast</td>
<td>81</td>
<td>75</td>
<td>true</td>
</tr>
<tr>
<td>14</td>
<td>yes</td>
<td>0.453</td>
<td>0.547</td>
<td>yes</td>
<td>rain</td>
<td>71</td>
<td>80</td>
<td>true</td>
</tr>
</tbody>
</table>

- **Classifier is quite sure**
- **Classifier is not sure**
Decision Boundary of Naive Bayes Classifier

- Usually larger coherent areas
- Soft margins with uncertain regions
- Arbitrary (often curved) shapes
Naïve Bayes (Summary)

- Robust to isolated noise points
  - they have a small impact on the probabilities

- 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)
Why Naïve Bayes?

• Recap:
  – we assume that all the attributes are independent

• This does not hold for many real world datasets
  – e.g., persons: sex, weight, height
  – e.g., cars: weight, fuel consumption
  – e.g., countries: population, area, GDP
  – e.g., food: ingredients
  – e.g., text: word occurrences ("Barack", "Obama")
  – ...

Naïve Bayes Discussion

• Naïve Bayes works surprisingly well.
  – even if independence assumption is clearly violated
  – Classification doesn’t require accurate probability estimates as long as maximum probability is assigned to correct class

• However: Adding too many redundant attributes will cause problems
  – Solution: Select attribute subset as Naïve Bayes often works as well or better with just a fraction of all attributes.

• Technical advantages:
  – Learning Naïve Bayes classifiers is computationally cheap as probabilities can be estimated doing one pass over the training data
  – Storing the probabilities does not require a lot of memory
Comparison kNN and Naïve Bayes

• Computation
  – Naïve Bayes is often faster

• Naïve Bayes uses *all* data points
  – less sensitive to local outliers

• Irrelevant attributes
  – are not problematic for Naïve Bayes
  – attribute values equally distributed across classes
    → same factor for each class
  – may be problematic for kNN

• In both cases
  – attribute pre-selection makes sense (see Data Mining II)
Lazy vs. Eager Learning

- Both k NN and Naïve Bayes are “lazy” methods
- They do not build an explicit model!
  - “learning” is only performed on demand for unseen records

```
<table>
<thead>
<tr>
<th>Tid</th>
<th>Attrib1</th>
<th>Attrib2</th>
<th>Attrib3</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Yes</td>
<td>Large</td>
<td>125K</td>
<td>No</td>
</tr>
<tr>
<td>2</td>
<td>No</td>
<td>Medium</td>
<td>100K</td>
<td>No</td>
</tr>
<tr>
<td>3</td>
<td>No</td>
<td>Small</td>
<td>70K</td>
<td>No</td>
</tr>
<tr>
<td>4</td>
<td>Yes</td>
<td>Medium</td>
<td>120K</td>
<td>No</td>
</tr>
<tr>
<td>5</td>
<td>No</td>
<td>Large</td>
<td>95K</td>
<td>Yes</td>
</tr>
<tr>
<td>6</td>
<td>No</td>
<td>Medium</td>
<td>80K</td>
<td>No</td>
</tr>
<tr>
<td>7</td>
<td>Yes</td>
<td>Large</td>
<td>220K</td>
<td>No</td>
</tr>
<tr>
<td>8</td>
<td>No</td>
<td>Small</td>
<td>85K</td>
<td>Yes</td>
</tr>
<tr>
<td>9</td>
<td>No</td>
<td>Medium</td>
<td>75K</td>
<td>No</td>
</tr>
<tr>
<td>10</td>
<td>No</td>
<td>Small</td>
<td>90K</td>
<td>Yes</td>
</tr>
<tr>
<td>11</td>
<td>No</td>
<td>Small</td>
<td>55K</td>
<td>?</td>
</tr>
<tr>
<td>12</td>
<td>Yes</td>
<td>Medium</td>
<td>80K</td>
<td>?</td>
</tr>
<tr>
<td>13</td>
<td>Yes</td>
<td>Large</td>
<td>110K</td>
<td>?</td>
</tr>
<tr>
<td>14</td>
<td>No</td>
<td>Small</td>
<td>95K</td>
<td>?</td>
</tr>
<tr>
<td>15</td>
<td>No</td>
<td>Large</td>
<td>67K</td>
<td>?</td>
</tr>
</tbody>
</table>
```
Lazy vs. Eager Learning

- We have seen the two most common techniques for lazy learning
  - $k$ nearest neighbors
  - Naïve Bayes

- We will see eager learning in the next lectures
  - where explicit models are built
  - e.g., decision trees
  - e.g., rule sets
Questions?