Data mining,
Data analysis, Statistical analysis,
Pattern discovery, Statistical learning,
Machine learning, Predictive analytics,
Business intelligence, Data-driven statistics
Inductive reasoning, Pattern analysis,
Knowledge discovery from databases,
Analytical processing,
...

Machine Learning :: Introduction
Part II

Konstantin Tretyakov (kt@ut.ee)

MTAT.03.183 – Data Mining
November 5, 2009
In the previous episode

Approaches to data analysis

The general principle is the same, though:
1. Define a **set of patterns** of interest
2. Define a **measure of goodness** for the patterns
3. Find the **best pattern** in the data
In the previous episode

Supervised Learning: Manual
In the previous episode

Supervised Learning: Automatic

Regression

Classification
In the previous episode

Supervised Learning

Regression

Classification

Validation

Training set

Test set

Holdout

Cross-validation

Accuracy, Precision, Recall

TP, FP, TN, FN
In the previous episode

Supervised Learning

Regression

Classification

Decision trees: ID3, C4.5
Today

Supervised Learning

Regression

Linear regression

Classification

Naïve Bayes

KNN
Supervised learning

- Four examples of approaches
  - Ad-hoc
    - Decision tree induction
  - Probabilistic modeling
    - Naïve Bayes classifier
  - Objective function optimization
    - Linear least squares regression
  - Instance-based methods
    - K-nearest neighbors
Naïve Bayes Classifier

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

<table>
<thead>
<tr>
<th>Day</th>
<th>Outlook</th>
<th>Temp</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>
Shall we play tennis today?

<table>
<thead>
<tr>
<th>PlayTennis</th>
</tr>
</thead>
<tbody>
<tr>
<td>No</td>
</tr>
<tr>
<td>No</td>
</tr>
<tr>
<td>Yes</td>
</tr>
<tr>
<td>Yes</td>
</tr>
<tr>
<td>Yes</td>
</tr>
<tr>
<td>No</td>
</tr>
<tr>
<td>Yes</td>
</tr>
<tr>
<td>Yes</td>
</tr>
<tr>
<td>Yes</td>
</tr>
<tr>
<td>Yes</td>
</tr>
<tr>
<td>Yes</td>
</tr>
<tr>
<td>No</td>
</tr>
</tbody>
</table>
Shall we play tennis today?

Probabilistic model:

<table>
<thead>
<tr>
<th>PlayTennis</th>
</tr>
</thead>
<tbody>
<tr>
<td>No</td>
</tr>
<tr>
<td>No</td>
</tr>
<tr>
<td>Yes</td>
</tr>
<tr>
<td>Yes</td>
</tr>
<tr>
<td>Yes</td>
</tr>
<tr>
<td>No</td>
</tr>
<tr>
<td>Yes</td>
</tr>
<tr>
<td>Yes</td>
</tr>
<tr>
<td>Yes</td>
</tr>
<tr>
<td>Yes</td>
</tr>
</tbody>
</table>

P(Yes) = 9/14 = 0.64
P(No) = 5/14 = 0.36

⇒ Yes
It’s windy today. Tennis, anyone?

<table>
<thead>
<tr>
<th>Wind</th>
<th>PlayTennis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weak</td>
<td>No</td>
</tr>
<tr>
<td>Strong</td>
<td>No</td>
</tr>
<tr>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>Strong</td>
<td>No</td>
</tr>
<tr>
<td>Strong</td>
<td>Yes</td>
</tr>
<tr>
<td>Weak</td>
<td>No</td>
</tr>
<tr>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>Strong</td>
<td>Yes</td>
</tr>
<tr>
<td>Strong</td>
<td>Yes</td>
</tr>
<tr>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>Strong</td>
<td>No</td>
</tr>
</tbody>
</table>
It’s windy today. Tennis, anyone?

- **Probabilistic model:**

<table>
<thead>
<tr>
<th>Wind</th>
<th>PlayTennis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weak</td>
<td>No</td>
</tr>
<tr>
<td>Strong</td>
<td>No</td>
</tr>
<tr>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>Strong</td>
<td>No</td>
</tr>
<tr>
<td>Strong</td>
<td>Yes</td>
</tr>
<tr>
<td>Weak</td>
<td>No</td>
</tr>
<tr>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>Strong</td>
<td>Yes</td>
</tr>
<tr>
<td>Strong</td>
<td>Yes</td>
</tr>
<tr>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>Strong</td>
<td>No</td>
</tr>
</tbody>
</table>

\[P(\text{Weak}) = \frac{8}{14}\]
\[P(\text{Strong}) = \frac{6}{14}\]
\[P(\text{Yes} \mid \text{Weak}) = \frac{6}{8}\]
\[P(\text{No} \mid \text{Weak}) = \frac{2}{8}\]
\[P(\text{Yes} \mid \text{Strong}) = \frac{3}{6}\]
\[P(\text{No} \mid \text{Strong}) = \frac{3}{6}\]
More attributes

> Probabilistic model:

<table>
<thead>
<tr>
<th>Humidity</th>
<th>Wind</th>
<th>PlayTennis</th>
</tr>
</thead>
<tbody>
<tr>
<td>High</td>
<td>Weak</td>
<td>No</td>
</tr>
<tr>
<td>High</td>
<td>Strong</td>
<td>No</td>
</tr>
<tr>
<td>High</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>High</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>Normal</td>
<td>Strong</td>
<td>No</td>
</tr>
<tr>
<td>Normal</td>
<td>Strong</td>
<td>Yes</td>
</tr>
<tr>
<td>High</td>
<td>Weak</td>
<td>No</td>
</tr>
<tr>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>Normal</td>
<td>Strong</td>
<td>Yes</td>
</tr>
<tr>
<td>High</td>
<td>Strong</td>
<td>Yes</td>
</tr>
<tr>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>High</td>
<td>Strong</td>
<td>No</td>
</tr>
</tbody>
</table>

\[
P(\text{High,Weak}) = \frac{4}{14}
\]
\[
P(\text{Yes} \mid \text{High,Weak}) = \frac{2}{4}
\]
\[
P(\text{No} \mid \text{High,Weak}) = \frac{2}{4}
\]
\[
P(\text{High,Strong}) = \frac{3}{14}
\]
\[
P(\text{Yes} \mid \text{High,Strong}) = \frac{1}{3}
\]
\[
P(\text{No} \mid \text{High,Strong}) = \frac{2}{3}
\]

...
The Bayes Classifier

In general:

1. **Estimate from data:**
   \[ P(\text{Class} \mid X_1, X_2, X_3, \ldots) \]

2. **For a given instance** \((X_1, X_2, X_3, \ldots)\)
   predict class whose conditional probability is **greater**:

   \[ P(C_1 \mid X_1, X_2, X_3, \ldots) > P(C_2 \mid X_1, X_2, X_3, \ldots) \]
   \(\Rightarrow\) predict \(C_1\)
The Bayes Classifier

In general:

1. \( P(C_1 | X) > P(C_2 | X) \)  
   \( \rightarrow \) predict \( C_1 \)
The Bayes Classifier

In general:

1. \( P(C_1 | X) > P(C_2 | X) \)  
   - predict \( C_1 \)

2. \( \frac{P(X | C_1)P(C_1)}{P(X)} > \frac{P(X | C_2)P(C_2)}{P(X)} \)  
   - predict \( C_1 \)
The Bayes Classifier

In general:

1. \( P(C_1 \mid X) > P(C_2 \mid X) \)  
   \( \Rightarrow \) predict \( C_1 \)

2. \( P(X \mid C_1)P(C_1)/P(X) > P(X \mid C_2)P(C_2)/P(X) \)  
   \( \Rightarrow \) predict \( C_1 \)

3. \( P(X \mid C_1)P(C_1) > P(X \mid C_2)P(C_2) \)  
   \( \Rightarrow \) predict \( C_1 \)
The Bayes Classifier

In general:

1. \( P(C_1 | X) > P(C_2 | X) \)
   \( \Rightarrow \) predict \( C_1 \)

2. \( P(X | C_1)P(C_1)/P(X) > P(X | C_2)P(C_2)/P(X) \)
   \( \Rightarrow \) predict \( C_1 \)

3. \( P(X | C_1)P(C_1) > P(X | C_2)P(C_2) \)
   \( \Rightarrow \) predict \( C_1 \)

4. \( P(X | C_1)/P(X | C_2) > P(C_2)/P(C_1) \)
   \( \Rightarrow \) predict \( C_1 \)
The Bayes Classifier

3. \[ P(X \mid C_1)P(C_1) > P(X \mid C_2)P(C_2) \implies \text{predict } C_1 \]
The Bayes Classifier

If the true underlying distribution is known, the Bayes classifier is optimal (i.e. it achieves minimal error probability).
We need exponential amount of data

<table>
<thead>
<tr>
<th>Humidity</th>
<th>Wind</th>
<th>PlayTennis</th>
</tr>
</thead>
<tbody>
<tr>
<td>High</td>
<td>Weak</td>
<td>No</td>
</tr>
<tr>
<td>High</td>
<td>Strong</td>
<td>No</td>
</tr>
<tr>
<td>High</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>High</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>Normal</td>
<td>Strong</td>
<td>No</td>
</tr>
<tr>
<td>Normal</td>
<td>Strong</td>
<td>Yes</td>
</tr>
<tr>
<td>High</td>
<td>Weak</td>
<td>No</td>
</tr>
<tr>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>Normal</td>
<td>Strong</td>
<td>Yes</td>
</tr>
<tr>
<td>High</td>
<td>Strong</td>
<td>Yes</td>
</tr>
<tr>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>High</td>
<td>Strong</td>
<td>No</td>
</tr>
</tbody>
</table>

\[
P(\text{High,Weak}) = \frac{4}{14}
\]

\[
P(\text{Yes | High,Weak}) = \frac{2}{4}
\]

\[
P(\text{No | High,Weak}) = \frac{2}{4}
\]

\[
P(\text{High,Strong}) = \frac{3}{14}
\]

\[
P(\text{Yes | High,Strong}) = \frac{1}{3}
\]

\[
P(\text{No | High,Strong}) = \frac{2}{3}
\]

...
The Naïve Bayes Classifier

To scale beyond 2-3 attributes, use a trick:

Assume that attributes of each class are independent:

\[
P(X_1, X_2, X_3 | \text{Class}) = P(X_1 | \text{Class})P(X_2 | \text{Class})P(X_3 | \text{Class}) \ldots
\]
The Naïve Bayes Classifier

\[ \frac{P(X | C_1)}{P(X | C_2)} > \frac{P(C_2)}{P(C_1)} \rightarrow \text{predict } C_1 \]

becomes

\[ \prod_{i} \frac{P(X_i | C_1)}{P(X_i | C_2)} > \frac{P(C_2)}{P(C_1)} \Rightarrow \text{Predict } C_1 \]
Naïve Bayes Classifier

1. **Training:**
   For each value $v$ of each attribute $i$, compute
   
   $$m(i, v) = \frac{P(X_i = v | C_1)}{P(X_i = v | C_2)}$$

2. **Classification:**
   For a given instance $(x_1, x_2, \ldots)$ compute
   
   $$\text{score} = \prod_{i} m(i, x_i)$$
   
   If $\frac{\text{score} \cdot P(C_2)}{P(C_1)} \geq 1$ classify as $C_1$, else $C_2$
Naïve Bayes Classifier

1. **Training:**
   For each value \( v \) of each attribute \( i \), compute
   
   \[
   l(i, v) = \log \frac{P(X_i = v | C_1)}{P(X_i = v | C_2)}
   \]

2. **Classification:**
   For a given instance \((x_1, x_2, \ldots)\) compute
   
   \[
   \text{lscore} = \sum_i l(i, x_i)
   \]
   
   If \( \text{lscore} > \log \frac{P(C_2)}{P(C_1)} \) classify as \( C_1 \), else \( C_2 \)
Naïve Bayes Classifier

- Extendable to continuous attributes via kernel density estimation.
- The goods:
  - Easy to implement, efficient
  - Won’t overfit, interpretable
  - Works better than you would expect (e.g. spam filtering)
- The bads
  - “Naïve”, linear
  - Usually won’t work well for too many classes
Supervised learning

- Four examples of approaches
  - Ad-hoc
    - Decision tree induction
  - Probabilistic modeling
    - Naïve Bayes classifier
  - Objective function optimization
    - Linear least squares regression
  - Instance-based methods
    - $K$-nearest neighbors
Linear regression

- Consider data points of the type 
  \[(x_i, y_i) \in \mathbb{R}\]

- Let us search for a function of the form 
  \[f(x) = w x\]
  that would have \textbf{the least possible sum of error squares:}
  \[E(w) = \sum_{i}(f(x_i) - y_i)^2\]
Linear regression

- Set of functions $f(x) = wx$ for various $w$. 
Linear regression

- The function having the least error
Linear regression

- We need to solve: \( \arg \min_w E(w) \)

where

\[ E(w) = \sum_i (wx_i - y_i)^2 \]

- Setting the derivative to 0:

\[ E'(w) = \sum_i 2(wx_i - y_i)x_i = 0 \]

we get

\[ w = \frac{\sum_i y_i x_i}{\sum_i x_i^2} \]
Linear regression

- The idea naturally generalizes to more complex cases (e.g. multivariate regression)

- The goods
  - Simple, easy
  - Interpretable, popular, extendable to many variants
  - Won’t overfit

- The bads
  - Linear, i.e. works for a limited set of datasets.
  - Nearly never perfect.
Supervised learning

Four examples of approaches

- Ad-hoc
  - Decision tree induction
- Probabilistic modeling
  - Naïve Bayes classifier
- Objective function optimization
  - Linear least squares regression
- Instance-based methods
  - K-nearest neighbors
K-Nearest Neighbors
K-Nearest Neighbors

- One-nearest neighbor
K-Nearest Neighbors

- One-nearest neighbor
K-Nearest Neighbors

**Training:**
- Store and index all training instances

**Classification**
- Given an instance to classify, find $k$ instances from the training sample nearest to it.
- Predict the majority class among these $k$ instances
K-Nearest Neighbors

- **The goods**
  - Trivial concept
  - Easy to implement
  - Asymptotically optimal
  - Allows various kinds of data

- **The bads**
  - Difficult to implement efficiently
  - Not interpretable (no model)
  - On smaller datasets loses to other methods
Supervised learning

**Ad-hoc**
- Decision trees, forests
- Rule induction, ILP
- Fuzzy reasoning

\[ H(p) = - \sum_i p_i \log_2 p_i \]

**Objective optimization**
- Regression models
- Kernel methods, SVM, RBF
- Neural networks

\[ \arg \min_w E(w) \]

**Probabilistic models**
- Naïve Bayes
- Graphical models
- Regression models
- Density estimation

\[ \prod_i \frac{P(X_i | C_1)}{P(X_i | C_2)} \geq \frac{P(C_2)}{P(C_1)} \]

**Instance-based**
- K-NN, LOWESS
- Kernel densities
- SVM, RBF

...
Supervised learning

Ad-hoc
- Decision trees, forests
- Rule induction, ILP
- Fuzzy reasoning

Objective optimization
- Regression models
- Kernel methods, SVM, RBF
- Neural networks
- \[ \arg \min_w E(w) \]

Probabilistic models
- Naïve Bayes
- Graphical models
- \[ \prod_{i} \frac{P(X_i|C_1)}{P(X_i|C_2)} \leq \frac{P(C_1)}{P(C_2)} \]

Instance-based
- K-NN, LOWESS
- Kernel densities
- SVM, RBF

Ensemble-learners
- Arcing, Boosting, Bagging, Dagging, Voting, Stacking

Machine learning :: Introduction :: Part II  12.11.2009
Coming up next

- “Machine learning”
  - Terminology, foundations, general framework.
- Supervised machine learning
  - Basic ideas, algorithms & toy examples.
- **Statistical challenges**
  - Learning theory, consistency, bias-variance, overfitting, ...
- State of the art techniques
  - SVM, kernel methods, graphical models, latent variable models, boosting, bagging, LASSO, on-line learning, deep learning, reinforcement learning, …
Questions?

Why are we doing this?
What problem are we solving?
Is this actually useful?
Are we adding value?
Will this change behavior?
Is there an easier way?
What's the opportunity cost?
Is it really worth it?