Data Mining

Classification

- Part 3 -
Outline

1. What is Classification?
2. K-Nearest-Neighbors
3. Decision Trees
4. Model Evaluation
5. Rule Learning
6. Naïve Bayes
7. Artificial Neural Networks
8. Support Vector Machines
9. Parameter Tuning
6. Naïve Bayes

- Probabilistic classification technique based on Bayes theorem.
  - Widely used and especially successful at classifying texts
- Goal: Estimate the most probable class label for a given record.
- Probabilistic formulation of the classification task:
  - Consider each attribute and class label as random variables
  - Given a record with attributes \((A_1, A_2, \ldots, A_n)\), the goal is to find the class \(C\) that maximizes the conditional probability

\[ P(C | A_1, A_2, \ldots, A_n) \]

- Example: Should we play golf?
  - \(P(\text{Play}=\text{yes} | \text{Outlook}=\text{rainy}, \text{Temperature}=\text{cool})\)
  - \(P(\text{Play}=\text{no} | \text{Outlook}=\text{rainy}, \text{Temperature}=\text{cool})\)
- Question: How to estimate these probabilities given training data?
Bayes Theorem

- Thomas Bayes (1701-1761)
  - British mathematician and priest
  - tried to formally prove the existence of God
- Bayes Theorem

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

- useful in situations where \( P(C|A) \) is unknown while \( P(A|C), P(A) \) and \( P(C) \) are known or easy to estimate
Bayes Theorem: Evidence Formulation

- **Prior probability** of event $H$:
  - Probability of event *before* evidence is seen.
  - We play golf in 70% of all cases $\Rightarrow P(H) = 0.7$

- **Posterior probability** of event $H$:
  - Probability of event *after* evidence is seen.
  - Evidence: It is windy and raining $\Rightarrow P(H \mid E) = 0.2$

- Probability of event $H$ given evidence $E$:

\[
P(H \mid E) = \frac{P(E \mid H)P(H)}{P(E)}
\]
Applying Bayes Theorem to the Classification Task

1. Compute the probability $P(C \mid A)$ for all values of $C$ using Bayes theorem.
   • $P(A)$ is the same for all classes. Thus, we need to estimate $P(C)$ and $P(A \mid C)$

2. Choose value of $C$ that maximizes $P(C \mid A)$.

Example:

$P(Play = yes \mid Outlook = rainy, Temp = cool) = \frac{P(Outlook = rainy, Temp = cool \mid Play = yes)P(Play = yes)}{P(Outlook = rainy, Temp = cool)}$

$P(Play = no \mid Outlook = rainy, Temp = cool) = \frac{P(Outlook = rainy, Temp = cool \mid Play = no)P(Play = no)}{P(Outlook = rainy, Temp = cool)}$
The prior probability $P(C_j)$ for each class is estimated by:

1. counting the records in the training set that are labeled with class $C_j$
2. dividing the count by the overall number of records

Example:
- $P(\text{Play}=\text{no}) = 5/14$
- $P(\text{Play}=\text{yes}) = 9/14$
Naïve Bayes assumes that all attributes are *statistically independent*. Knowing the value of one attribute says nothing about the value of another. This independence assumption is almost never correct! But … this scheme works well in practice.

The independence assumption allows the joint probability $P(A | C)$ to be reformulated as the product of the individual probabilities $P(A_i | C_j)$:

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

$$P(\text{Outlook}=\text{rainy}, \text{Temperature}=\text{cool} | \text{Play}=\text{yes}) = P(\text{Outlook}=\text{rainy} | \text{Play}=\text{yes}) \times P(\text{Temperature}=\text{cool} | \text{Play}=\text{yes})$$

Result: The probabilities $P(A_i | C_j)$ for all $A_i$ and $C_j$ can be estimated directly from the training data.
Estimating the Probabilities $P(A_i | C_j)$

<table>
<thead>
<tr>
<th>Outlook</th>
<th>Temperature</th>
<th>Humidity</th>
<th>Windy</th>
<th>Play</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sunny</td>
<td>2/9</td>
<td>3/9</td>
<td>High</td>
<td>No</td>
</tr>
<tr>
<td>Overcast</td>
<td>4/9</td>
<td>0/5</td>
<td>Mild</td>
<td>Yes</td>
</tr>
<tr>
<td>Rainy</td>
<td>3/9</td>
<td>2/5</td>
<td>Cool</td>
<td>Yes</td>
</tr>
</tbody>
</table>

The probabilities $P(A_i | C_j)$ are estimated by:
1. counting how often an attribute value appears together with class $C_j$
2. dividing the count by the overall number of records belonging to class $C_j$

Example:
2 times “Yes” together with “Outlook=sunny” out of altogether 9 “Yes” examples
$\Rightarrow p(\text{Outlook}=\text{sunny}|\text{Yes}) = \frac{2}{9}$
Classifying a New Day

Unseen Record

<table>
<thead>
<tr>
<th>Outlook</th>
<th>Temp.</th>
<th>Humidity</th>
<th>Windy</th>
<th>Play</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sunny</td>
<td>Cool</td>
<td>High</td>
<td>True</td>
<td>?</td>
</tr>
</tbody>
</table>

\[
\Pr[yes \mid E] = \Pr[Outlook = Sunny \mid yes] \\
\times \Pr[Tempuratur = Cool \mid yes] \\
\times \Pr[Humidity = High \mid yes] \\
\times \Pr[Windy = True \mid yes] \\
\times \frac{\Pr[yes]}{\Pr[E]} \\
= \frac{2}{9} \times \frac{3}{9} \times \frac{3}{9} \times \frac{3}{9} \times \frac{9}{14} \frac{14}{Pr[E]}
\]

Probability of class “yes” given the evidence

Class-conditional probability of the evidence

Prior probability of class “yes”

Prior probability of evidence
Classifying a New Day: Weigh the evidence!

<table>
<thead>
<tr>
<th>Outlook</th>
<th>Temperature</th>
<th>Humidity</th>
<th>Windy</th>
<th>Play</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sunny</td>
<td>Yes: 2, No: 3</td>
<td>Hot: 2, No: 2</td>
<td>High: 3, No: 4</td>
<td>False: 6, Yes: 2</td>
</tr>
<tr>
<td>Overcast</td>
<td>Yes: 4, No: 0</td>
<td>Mild: 4, No: 2</td>
<td>Normal: 6, No: 1</td>
<td>True: 3, Yes: 3</td>
</tr>
<tr>
<td>Rainy</td>
<td>Yes: 3, No: 2</td>
<td>Cool: 3, No: 1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- A new day:

  Prior probability
  Evidence

  Choose Maximum

Likelihood of the two classes

- For “yes” = \( \frac{2}{9} \times \frac{3}{9} \times \frac{3}{9} \times \frac{3}{9} \times \frac{9}{14} = 0.0053 \)
- For “no” = \( \frac{3}{5} \times \frac{1}{5} \times \frac{4}{5} \times \frac{3}{5} \times \frac{5}{14} = 0.0206 \)

Conversion into a probability by normalization:

- \( P(“yes”) = \frac{0.0053}{0.0053 + 0.0206} = 0.205 \)
- \( P(“no”) = \frac{0.0206}{0.0053 + 0.0206} = 0.795 \)
Handling Numerical Attributes

- Option 1: **Discretize** numerical attributes before learning classifier.
  - Temp = 37°C ➔ “Hot”
  - Temp = 21°C ➔ “Mild”

- Option 2: Make assumption that numerical attributes have a **normal distribution** given the class.
  - Use training data to estimate parameters of the distribution (e.g., mean and standard deviation)
  - Once the probability distribution is known, it can be used to estimate the conditional probability $P(A_i|C_j)$
Handling Numerical Attributes

- The probability density function for the normal distribution is

\[ f(x) = \frac{1}{\sqrt{2\pi \sigma}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \]

- It is defined by two parameters:
  
  - **Sample mean** \( \mu \)
    \[ \mu = \frac{1}{n} \sum_{i=1}^{n} x_i \]
  
  - **Standard deviation** \( \sigma \)
    \[ \sigma = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \mu)^2 \]

- Both parameters can be estimated from the training data.
Statistics for the Weather Data

<table>
<thead>
<tr>
<th>Outlook</th>
<th>Temperature</th>
<th>Humidity</th>
<th>Windy</th>
<th>Play</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sunny</td>
<td>2/9</td>
<td>2/5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Overcast</td>
<td>2/9</td>
<td>0/5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rainy</td>
<td>3/9</td>
<td>2/5</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Example calculation:

\[ f (temp = 66 \mid yes) = \frac{1}{\sqrt{2\pi \times 6.2}} e^{-\frac{(66-73)^2}{2 \times 6.2^2}} = 0.0340 \]
Classifying a New Day

Unseen Record

<table>
<thead>
<tr>
<th>Outlook</th>
<th>Temp.</th>
<th>Humidity</th>
<th>Windy</th>
<th>Play</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sunny</td>
<td>66</td>
<td>90</td>
<td>true</td>
<td>?</td>
</tr>
</tbody>
</table>

Likelihood of “yes” = \( \frac{2}{9} \times 0.0340 \times 0.0221 \times \frac{3}{9} \times \frac{9}{14} = 0.000036 \)

Likelihood of “no” = \( \frac{3}{5} \times 0.0291 \times 0.0380 \times \frac{3}{5} \times \frac{5}{14} = 0.000136 \)

\[ P(“yes”) = \frac{0.000036}{0.000036 + 0.000136} = 20.9\% \]

\[ P(“no”) = \frac{0.000136}{0.000036 + 0.000136} = 79.1\% \]

But note: Some numeric attributes are not normally distributed and you may thus need to choose a different probability density function or use discretization.
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**:

<table>
<thead>
<tr>
<th>Outlook</th>
<th>Temp.</th>
<th>Humidity</th>
<th>Windy</th>
<th>Play</th>
</tr>
</thead>
<tbody>
<tr>
<td>?</td>
<td>Cool</td>
<td>High</td>
<td>True</td>
<td>?</td>
</tr>
</tbody>
</table>

Likelihood of “yes” = \( \frac{3}{9} \times \frac{3}{9} \times \frac{3}{9} \times \frac{9}{14} = 0.0238 \)

Likelihood of “no” = \( \frac{1}{5} \times \frac{4}{5} \times \frac{3}{5} \times \frac{5}{14} = 0.0343 \)

\[ P(“yes”) = \frac{0.0238}{0.0238 + 0.0343} = 41\% \]

\[ P(“no”) = \frac{0.0343}{0.0238 + 0.0343} = 59\% \]
The Zero-Frequency Problem

- What if an attribute value doesn’t occur with every class value? (e.g. no “Outlook = overcast” for class “no”)
  - Class-conditional probability will be zero! \( P[\text{Outlook} = \text{overcast} \mid \text{no}] = 0 \)
  - Problem: Posterior probability will also be zero! \( P[\text{no} \mid E] = 0 \) (No matter how likely the other values are!)

- Remedy: Add 1 to the count for every attribute value-class combination (Laplace Estimator)

- Result: Probabilities will never be zero! (also: stabilizes probability estimates)

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

Laplace: \( P(A_i \mid C) = \frac{N_{ic} + 1}{N_c + c} \) \( c: \) number of classes
Naïve Bayes in RapidMiner
Naïve Bayes in RapidMiner: The 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>0.438</td>
</tr>
<tr>
<td>Outlook</td>
<td>value=sunny</td>
<td>0.581</td>
<td>0.223</td>
</tr>
<tr>
<td>Outlook</td>
<td>value=unknown</td>
<td>0.014</td>
<td>0.008</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>
Naïve Bayes in RapidMiner: Confidence Scores

<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>true</td>
</tr>
<tr>
<td>6</td>
<td>no</td>
<td>0.336</td>
<td>0.664</td>
<td>no</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>true</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>false</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>true</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>yes</td>
<td>0.038</td>
<td>0.962</td>
<td>true</td>
<td></td>
<td></td>
<td></td>
<td></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
Characteristics of Naïve Bayes

- Naïve Bayes works surprisingly well for many classification tasks.
  - even if independence assumption is clearly violated
  - Why? Because classification doesn’t require accurate probability estimates as long as maximum probability is assigned to correct class

- Robust to isolated noise points as they will be averaged out
- Robust to irrelevant attributes as \( P(A_i|C) \) distributed uniformly for \( A_i \)
- Adding too many redundant attributes can 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 on memory.
Alternative Classification Methods

- There are various methods of classification
  - e.g., RapidMiner implements 53 different methods

- So far, we have seen
  1. k-NN
  2. Decision Trees
  3. C4.5 and Ripper
  4. Naive Bayes

- Now: Brief introduction to
  1. Artificial Neural Networks
  2. Support Vector Machines
7. Artificial Neural Networks (ANN)

• Inspiration
  – one of the most powerful super computers in the world
Artificial Neural Networks (ANN)

Example:
Output Y is 1 if at least two of the three inputs are equal to 1.
Artificial Neural Networks (ANN)

Training Data

<table>
<thead>
<tr>
<th>$X_1$</th>
<th>$X_2$</th>
<th>$X_3$</th>
<th>$Y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

\[ Y = I(0.3X_1 + 0.3X_2 + 0.3X_3 - 0.4 > 0) \]

where \( I(z) = \begin{cases} 1 & \text{if } z \text{ is true} \\ 0 & \text{otherwise} \end{cases} \)
Artificial Neural Networks (ANN)

- Model is an assembly of inter-connected nodes and weighted links.
- Output node sums up each of its input value according to the weights of its links.
- Classification decision: Compare output node against some threshold $t$.

Perceptron Model:

$$ Y = I\left(\sum_i w_i X_i - t\right) \quad \text{or} \quad Y = \text{sign}\left(\sum_i w_i X_i - t\right) $$
Training ANN means learning the weights of the neurons.
Algorithm for Training ANNs

1. Initialize the weights \((w_0, w_1, \ldots, w_k)\), e.g., all with 1

2. Adjust the weights in such a way that the output of ANN is as consistent as possible with class labels of the training examples

   - Objective function:
     \[
     E = \sum_i \left[ Y_i - f(w_i, X_i) \right]^2
     \]

   - Find the weights \(w_i\)'s that minimize the error \(E\)
   - using for example the back propagation algorithm
     (see Tan Steinbach, Chapter 5.4)
Artificial Neural Networks in RapidMiner
Characteristics of Artificial Neural Networks

- ANNs can be used for classification as well as numerical regression tasks
- Multi-layer neural networks are universal approximators
  - meaning that they can approximate any target function
- Very important but difficult to choose the right network topology
  - Expressive hypothesis space can lead to overfitting
  - Possible approach to deal with overfitting:
    1. Start with several hidden layers and larger number of nodes
    2. Estimate generalization error using validation dataset
    3. Step by step remove nodes as long as generalization error improves
8. Support Vector Machines

- Support vector machines (SVMs) are algorithms for learning linear classifiers for
  - two class problems (a positive and a negative class)
  - from examples described by continuous attributes.

- SVMs achieve very good results especially for high dimensional data.

- SVMs were invented by V. Vapnik and his co-workers in 1970s in Russia and became known to the West in 1992.
Support Vector Machines

- SVMs find a linear **hyperplane** (decision boundary) that will separate the data.
Support Vector Machines

- Which one is better? B1 or B2?
- How do you define “better”?
Which Hyperplane is better?

- In order to avoid overfitting and to generalize for unseen data, SVMs find the hyperplane that maximizes the margin to the closest points (support vectors).

- Visual solution:
  - B1 is better than B2

- Mathematical solution:
  - *Constrained optimization* that can be solved using quadratic programming.
  - See Tan/Steinbach/Kumar, Chapter 5.5
Dealing with Not Linearly Separable Data

- What if the problem is not linearly separable due to noise points?

- Solution: Introduce slack variables in margin computation which result in a penalty for each data point that violates decision boundary.
Dealing with Not Linear Decision Boundaries

- Problem: What if decision boundary is not linear?
- Solution: Transform data into higher dimensional space where there is a linear separation
  - Details: Higher mathematics (see Tan/Steinbach, Chapter 5.5)
  - Different types of kernel functions are used for this transformation
Characteristics of Support Vector Machines

- SVM are often the most successful classification technique for high dimensional data.

- Application areas of SVM include
  - Text classification
  - Machine vision, e.g. face identification
  - Handwritten digit recognition
  - SPAM detection
  - Bioinformatics
SVMs in RapidMiner

Tuning a SVM

1. Transform all attributes to numeric scale (Operator: Nominal to Numeric)
2. Normalize all value ranges to [0,1] (Operator: Normalize)
3. Use the RBF kernel function
4. Use cross-validation to find the best values for the parameters
   1. C = weight of slack variables (Range: 0.03 to 30000)
   2. gamma = kernel parameter (Range: 0.00003 to 8)

9. Parameter Tuning

- Many learning methods require parameters
  - k for k-nearest-neighbors
  - pruning thresholds for trees and rules
  - hidden layer configuration for ANN
  - gamma and C for SVM

- Some methods often work rather poorly with default parameters.

- How to determine the optimal parameters?
  - Play around with different parameters in RapidMiner yourself
  - Alternative: Let RapidMiner test different parameter settings for you
Parameter Optimization

The **Optimize Parameters** operator allows you to automatically test various parameter combinations.
Parameter Optimization

- List of operators
- Parameters of selected operator
- Parameters to optimize
- Definition of parameter values for testing
- Final number of combinations!
- Steps linear/logarithmic (log good for SVMs)

Alternative optimization approaches that are more efficient than brute force grid search are beam search and evolutionary algorithms.
Attribute Subset Selection

The **Optimize Selection** operator allows you to find the optimal subset of the available attributes.

Forward selection: Find best single attribute, add further attributes, test again
Backward selection: Start with all attribute, remove attributes, test again
Outputing Interim Results

- Using the Log Operator

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Operator</th>
<th>Parameter or port</th>
<th>Selection of output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shuffle</td>
<td>Neural Net</td>
<td>parameter</td>
<td>shuffle</td>
</tr>
<tr>
<td>Training</td>
<td>Neural Net</td>
<td>parameter</td>
<td>training_cycles</td>
</tr>
<tr>
<td>Accuracy</td>
<td>Validation</td>
<td>value</td>
<td>performance</td>
</tr>
</tbody>
</table>

**Chapter 5.3: Naïve Bayes**

**Chapter 5.4: Artificial Neural Networks**

**Chapter 5.5: Support Vector Machines**

**Videos**

- Parameter Optimization
  
  http://www.youtube.com/watch?v=R5vPrTLMzng

- Attribute Subset Selection
  - Part 1: http://www.youtube.com/watch?v=7IC3IQEdWxA
  - Part 2: http://www.youtube.com/watch?v=j5vhwbLIZWg