

SEMINAR AI-TOOLS  
WS 2006/07

AUSGEWÄHLTE ANWENDUNGEN VON  
WEKA/MACHINE LEARNING

STEFAN WARWAS  
Waldhausweg 15  
66123 Saarbrücken  
stefan.warwas@freenet.de

14. APRIL 2007

BETREUER:  
MICHAEL FELD

UNIVERSITÄT DES SAARLANDES  
FACHBEREICH INFORMATIK

## **Zusammenfassung**

Machine-Learning Algorithmen kommen heute immer häufiger in praktischen Anwendungen zum Einsatz. In den meisten Fällen wird dabei auf Standard-Algorithmen zurückgegriffen. Weka ist eine in Java implementierte Machine-Learning Arbeitsumgebung, die bereits über eine große Anzahl von implementierten Machine-Learning Algorithmen verfügt. Darüber hinaus gibt Weka dem Benutzer einige interessante Tools an die Hand, die ihn bei der Datenanalyse unterstützen. In dieser Arbeit wird anhand von zwei Anwendungsbeispielen eine Fallstudie durchgeführt. Ziel ist es zu zeigen, wie die in der Literatur genannten Probleme mit Hilfe von Weka umgesetzt werden können.

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>2</b>
1.1	Anwendungsfall aus der Landwirtschaft . . . . .	2
1.2	Music Information Retrieval . . . . .	3
<b>2</b>	<b>Machine-Learning Prozessmodell</b>	<b>4</b>
2.1	Beteiligte Parteien . . . . .	5
2.2	Vorverarbeitung . . . . .	5
2.3	Attributanalyse . . . . .	6
2.4	Anwendung der Machine-Learning Algorithmen . . . . .	6
<b>3</b>	<b>Weka Arbeitsumgebung</b>	<b>7</b>
<b>4</b>	<b>Fallstudie</b>	<b>11</b>
4.1	Anwendungsfall aus der Landwirtschaft . . . . .	11
4.1.1	Vorverarbeitung . . . . .	12
4.1.2	Attributanalyse . . . . .	14
4.1.3	Anwendung der Machine-Learning Algorithmen . . . . .	14
4.2	Music Information Retrieval . . . . .	16
4.2.1	Vorverarbeitung . . . . .	17
4.2.2	Attributanalyse . . . . .	17
4.2.3	Anwendung der Machine-Learning Algorithmen . . . . .	18
<b>5</b>	<b>Zusammenfassung</b>	<b>19</b>
	<b>Literaturverzeichnis</b>	<b>20</b>

# Kapitel 1

## Einleitung

Machine-Learning (ML) kommt heute immer häufiger in praktischen Anwendungen zum Einsatz. Beispiele hierfür sind Spam-Filter (z.B. [SPAM]), Robotertechnik (z.B. [ROBOT]) oder auch Data-Mining (z.B. [GRO99]). Um Machine-Learning Algorithmen auch in eigenen Anwendungen und Experimenten zu verwenden, müssen die Standard-Algorithmen nicht jedes Mal neu implementiert werden.

Unter dem Namen *Waikato Environment for Knowledge Analysis* (Weka) entwickelt die neuseeländische *University of Waikato* seit 1993 eine Machine-Learning Arbeitsumgebung, die besonders für den praktischen Einsatz in Projekten geeignet ist. Weka bietet neben einer Java Programmier-Schnittstelle (API), einer Kommandozeilen-Schnittstelle (CLI) auch eine komfortable graphische Benutzer-Schnittstelle (GUI) um die Anforderungen der unterschiedlichen Nutzergruppen bestmöglich zu erfüllen. Der Schwerpunkt von Weka liegt dabei auf Klassifikationsalgorithmen. Außerdem wurden auch Algorithmen für *Regression*, *Clustering* und *Association Rules* implementiert. In dieser Seminararbeit wird eine Fallstudie anhand von zwei praktischen Anwendungsbeispielen durchgeführt. Es wird erläutert, wie die Funktionalität der Weka Machine-Learning Arbeitsumgebung eingesetzt werden kann, um die jeweiligen Problemstellungen zu lösen.

In Kapitel 1.1 und 1.2 werden zunächst die beiden Anwendungsbeispiele kurz vorgestellt. Anschließend wird in Kapitel 2 das Prozessmodell, auf dem Weka basiert, erläutert. Kapitel 3 gibt eine allgemeine Übersicht über Weka. In Kapitel 4 wird anschließend die Fallstudie anhand der beiden Anwendungsbeispiele durchgeführt. Kapitel 5 fasst diese Ausarbeitung zusammen.

### 1.1 Anwendungsfall aus der Landwirtschaft

Im ersten Anwendungsfall geht es um die Auswertung von Daten, die von neuseeländischen Milchkühen stammen. Dieses Anwendungsbeispiel wurde aus [GAR95] entnommen.

Im Durchschnitt werden in Neuseeland jährlich 20% aller Milchkühe geschlachtet. Für die Milchbauern stellt sich dabei die Frage welche ihrer Tiere sie schlachten und welche sie behalten. Um diese Entscheidung zu erleichtern sammelt die neuseeländische *Livestock Corporation* die von den Bauern aufgenommenen Daten der Milchkühe und berechnet daraus verschiedene Kennwerte, die den Milchkuhbauern die Selektion erleichtern.

Um die Selektion mit Hilfe von Machine-Learning Algorithmen zu verbessern haben sich die *Livestock Corporation* und die *University of Waikato* zusammengeschlossen. Dazu wurden der *University of Waikato* Daten von zehn Herden über sechs Jahre von der *Livestock Corporation* zur Verfügung gestellt.

Weitere Informationen zum Landwirtschaftsbeispiel befinden sich in Abschnitt 4.1.

## 1.2 Music Information Retrieval

Das zweite Anwendungsbeispiel stammt aus dem Themengebiet *Music Information Retrieval* (MIR). Unter *Music Information Retrieval* versteht man das Extrahieren von Information aus Musik mit Hilfe von Machine-Learning Algorithmen. Zum einen werden die Informationen direkt aus den Audiodaten extrahiert, zum anderen können auch externe Daten, wie zum Beispiel Liedtexte oder zusätzliche Metainformationen, herangezogen werden. Beispiele für die Anwendung von *Music Information Retrieval* sind die automatische Klassifikation von Liedern in Genres (z.B. [TZA01]) oder die Ähnlichkeitsbestimmung von Liedern (z.B. [AUC02]).

Das Anwendungsbeispiel für diese Ausarbeitung wurde aus [POH05] entnommen. In dieser Untersuchung wurden alle Kombinationen aus vier verschiedenen Merkmalsätzen und zwölf unterschiedlichen Machine-Learning Algorithmen daraufhin evaluiert, wie gut sie Lieder in sechs unterschiedliche Wahrnehmungskategorien klassifizieren. Um die beste Kombination aus einem Machine-Learning Algorithmus und einem Merkmalsatz für jede der sechs Wahrnehmungskategorien herauszufinden, wurden alle möglichen Kombinationen (ML Algorithmus \* Merkmalsatz \* Wahrnehmungskategorie) getestet.

Weitere Details zu diesem Anwendungsbeispiel befinden sich in Abschnitt 4.2.

Bevor die eigentliche Fallstudie durchgeführt wird, werden in den nächsten beiden Kapiteln die Weka Arbeitsumgebung und das ihr zugrunde liegende Prozessmodell vorgestellt.



## 2.1 Beteiligte Parteien

An einem aus der Praxis stammenden Machine-Learning Problem sind meist zwei Parteien beteiligt:

**Fachliche Experten** Fachliche Experten sind die Personen, die das zu lösende Problem definieren. Sie geben die Ziele vor, liefern die zu analysierenden Daten und besitzen das Wissen, um die Fragen der Machine-Learning Experten bezüglich der Problem-Domäne zu beantworten.

**Machine-Learning Experten** Machine-Learning Experten sind erfahren im Umgang mit Machine-Learning Algorithmen. Sie besitzen das Wissen um die für die Problemstellung passenden Algorithmen und deren Parameter optimal festzulegen. Bestehen Fragen bezüglich der zu untersuchenden Daten, so wenden sie sich an die fachlichen Experten.

Nur eine enge Zusammenarbeit beider Parteien führt zum gewünschten Ergebnis. Beim *Music Information Retrieval* Anwendungsbeispiel existiert diese Rollenverteilung nicht, da beide Rollen von der selben Person bzw. Gruppe wahrgenommen werden. Bei Problemen aus der Praxis existiert die oben genannte Rollenverteilung aber fast immer. So nehmen die Angestellten der *Livestock Corporation* im Landwirtschaftsbeispiel die Rolle der fachlichen Experten ein, und die Rolle der Machine-Learning Experten wird von den Mitarbeitern der *University of Waikato* wahrgenommen.

Der erste Schritt in einem Machine-Learning Projekt ist die Vorverarbeitung der Rohdaten (siehe Abbildung 2.1).

## 2.2 Vorverarbeitung

Zu Beginn werden die Rohdaten für die Analyse von den fachlichen Experten zur Verfügung gestellt. Üblicherweise enthalten Datensätze aus der Praxis viele Ungereimtheiten, die sie für eine direkte Verwendung, z.B. mit einem Machine-Learning Algorithmus, ungeeignet machen. Häufige Ursachen hierfür sind

- fehlende bzw. fehlerhafte Daten
- Rauschen in den Daten
- ungenaue bzw. unklare Bedeutung von Attributen
- ungünstig verteilte Daten (*Small Disjunct Problem*)

Während der Vorverarbeitung werden die Rohdaten zuerst importiert (z.B. in das ARFF Dateiformat [ARFF]). Alle Ungereimtheiten, die dabei

beobachtet werden, wie zum Beispiel fehlende Daten für ein Attribut in manchen Datensätzen, müssen mit den fachlichen Experten geklärt und anschließend ausgebessert werden. Das Ergebnis der Vorverarbeitung sind “saubere” und konsistente Daten. Der nächste Schritt besteht in der Auswahl der relevanten Merkmale (siehe Abbildung 2.1).

## 2.3 Attributanalyse

Bei Problemstellungen mit sehr vielen Datensätzen und Attributen kann es sein, dass manche Algorithmen ungeeignet sind, weil sie zu lange benötigen oder zu viel Speicher verbrauchen würden. Um dies zu vermeiden und den Lernprozess allgemein zu beschleunigen, versucht man mit Hilfe von statistischen Tests die Attribute auszuwählen, die für die jeweilige Problemstellung signifikant sind. Dadurch kann sich zum einen der Aufwand bei der Berechnung deutlich verringern. Zum anderen kann das Ergebnis selbst verbessert werden, wenn Attribute mit negativem Einfluß ausgefiltert werden.

Bei der Attributanalyse kann es außerdem Sinn machen, abgeleitete Attribute einzuführen. Dies ist beispielsweise dann von Vorteil, wenn die Bedeutung von existierenden Attributen unscharf ist und man mehrere Attribute zu einem aussagekräftigeren Attribut zusammenfassen möchte. Das Ergebnis der Attributanalyse ist die Auswahl der für die Problemstellung relevanten Merkmale.

## 2.4 Anwendung der Machine-Learning Algorithmen

Nachdem die Rohdaten vorverarbeitet und die signifikanten Attribute ausgewählt wurden, können nun Machine-Learning Algorithmen auf die Daten angewandt werden. Die gewonnenen Ergebnisse, wie zum Beispiel Klassifikationsbäume, müssen ausgewertet und anschließend mit den fachlichen Experten besprochen werden. Dabei stellt sich heraus, ob man neue Informationen gewinnen konnte oder ob noch Nachbesserungen nötig sind. Nach der einmaligen Anwendung von Machine-Learning Algorithmen ist allerdings noch nicht das Ende des Prozesses erreicht. Die gewonnenen Informationen können zum Beispiel dazu genutzt werden weitere abgeleitete Attribute einzuführen und so das Ergebnis zu verbessern. Allerdings muss darauf geachtet werden, dass dadurch kein *Bias* entsteht. Dieser Schritt wird so lange iterativ ausgeführt, bis das bestmögliche Ergebnis erreicht wurde (siehe Abbildung 2.1).

Nachdem in diesem Kapitel das Weka Prozessmodell erläutert wurde, wird im nächsten Kapitel die Weka Arbeitsumgebung selbst kurz vorgestellt.

## Kapitel 3

# Weka Arbeitsumgebung

Im vorangegangenen Kapitel wurden die unterschiedlichen Benutzergruppen vorgestellt. Während Machine-Learning Experten meist Kommandozeilen als Benutzer-Schnittstelle bevorzugen, ist für nicht-Informatiker eine graphische Benutzeroberfläche intuitiver und besser geeignet. Da unterschiedliche Benutzergruppen auch unterschiedliche Anforderungen an eine Machine-Learning Arbeitsumgebung stellen, bietet Weka zahlreiche Benutzer-Schnittstellen an. In diesem Kapitel werden nun die einzelnen Bestandteile bzw. Benutzeroberflächen von Weka kurz vorgestellt.

**Explorer** Der Weka Explorer ist ein Allzweckwerkzeug, das dem Benutzer über eine graphische Benutzeroberfläche erlaubt Daten

- zu importieren (z.B. von einer Datenbank oder einer Internetadresse in das ARFF Dateiformat),
- zu bearbeiten (z.B. Ändern einzelner Werte in den Daten, Anwenden von Filtern auf die Daten),
- zu visualisieren und schließlich Machine-Learning Algorithmen auf sie anzuwenden.

Außerdem bietet der Weka Explorer die Möglichkeit die signifikanten Attribute auszuwählen (z.B. mit *Forward Subset Selection* und *z-Score*). Abbildung 3.1 zeigt die Startseite des Explorers.

Der Weka Explorer ist hauptsächlich dazu gedacht, unbekannte Daten zu analysieren und die Parameter der Machine-Learning Algorithmen einzustellen. Weitere Informationen finden sich unter [WEKADOC].

**Experimenter** Der Weka Experimenter ist speziell für das Konfigurieren und Durchführen von Machine-Learning Experimenten entwickelt worden.

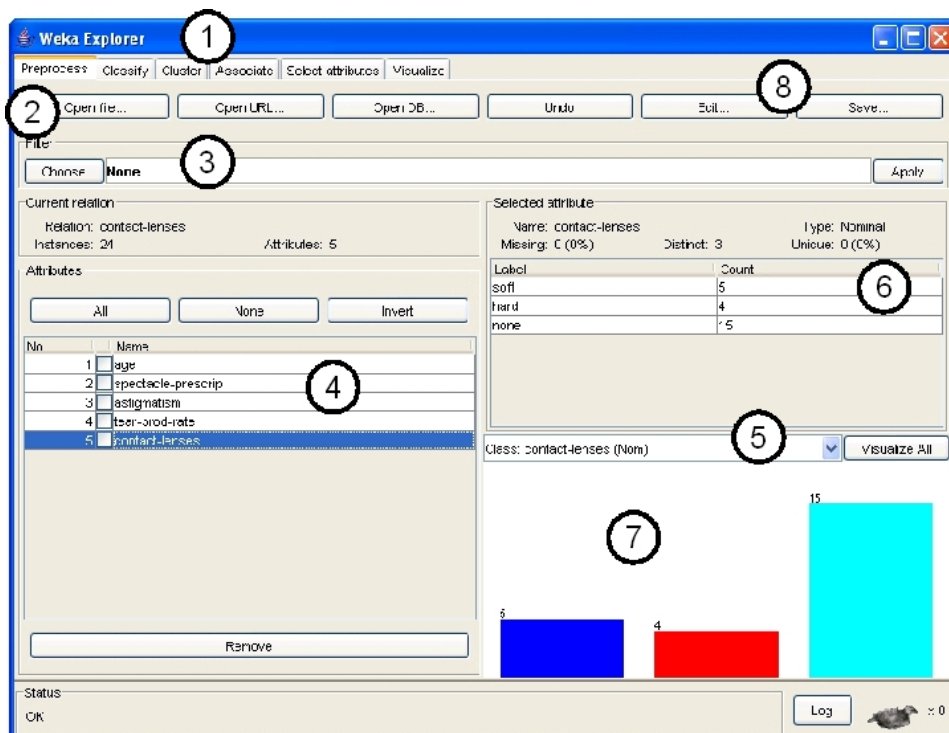


Abbildung 3.1: Weka Explorer Übersicht; 1: Auswahl der einzelnen Bereiche, 2: Importieren der Daten, 3: Auswahl von Filtern, 4: Auswahl von Attributen, auf die die Filter angewendet werden, 5: Wahl des Klassenattributs, 6: Kennwerte für aktuelles Attribut, 7: Visualisierung des gewählten Attributs, 8: Editieren der Daten.

Über eine graphische Benutzeroberfläche kann der Nutzer ein neues Experiment erstellen und konfigurieren. Dabei lassen sich unter anderem folgende Parameter festlegen:

- eine Datei, in die das Ergebnis geschrieben wird
- eine Liste von Machine-Learning Algorithmen, die angewendet werden sollen
- eine Liste von Eingabedateien, auf die die Machine-Learning Algorithmen angewendet werden sollen
- Typ des Experiments (z.B. “Cross Validation”)

Hat man ein Experiment vollständig konfiguriert, so kann man es automatisch ausführen lassen. Experimente lassen sich auch speichern und laden. Neben dem Standard-Experimenter existiert auch ein “Remote Experimenter”, der es erlaubt Experimente auf mehreren Rechnern parallel auszuführen. Hierfür muss auf jedem der Clients eine Weka “Remote Engine” installiert werden. Darüber hinaus wird auch ein zentraler Datenbankserver benötigt. Details finden sich unter [WEKADOC].

**Knowledge Flow** Die Weka Knowledge Flow Benutzeroberfläche ist eine Alternative zum Explorer und erlaubt es dem Nutzer, die Datenflüsse zwischen den einzelnen Komponenten, ähnlich wie in einem Datenflussdiagramm, zu modellieren (siehe Abbildung 3.2).

Beispielsweise ist es möglich, ein graphisches Symbol in das Diagramm zu setzen, das eine ARFF Eingabedatei repräsentiert (siehe Abbildung 3.2). Fügt man nun einen “NaiveBayesUpdateable Classifier” in das Diagramm ein und verbindet die beiden Elemente, so werden die Daten aus der ARFF Datei geladen und an den Klassifizierer weitergeleitet. Über die Optionen des Klassifizierers lässt sich einstellen, ob er von den Daten lernt oder sie klassifiziert. Weitere Informationen befinden sich unter [WEKADOC]. Mit Hilfe von Weka Knowledge Flow lässt sich der Machine-Learning Prozess sehr gut visualisieren. Weka Knowledge Flow ist im Moment noch in der Entwicklung und bietet deshalb noch nicht den gleichen Umfang wie die anderen Benutzer-Schnittstellen.

**Simple Command Line Interface** Über das Weka Simple Command Line Interface ist der volle Funktionsumfang von Weka über eine Textkonsole nutzbar. Öffnet man die Textkonsole, so lassen sich dort einfache Befehle, wie zum Beispiel “java weka.classifiers.trees.J48 -t weather.arff”, ausführen. Der eben genannte Befehl wendet einen “J48” Algorithmus (Wekas *C4.5* Implementierung) auf die Daten der Datei “weather.arff” an. Außerdem lässt sich der Weka Experimenter auch über die Textkonsole benutzen. Weitere

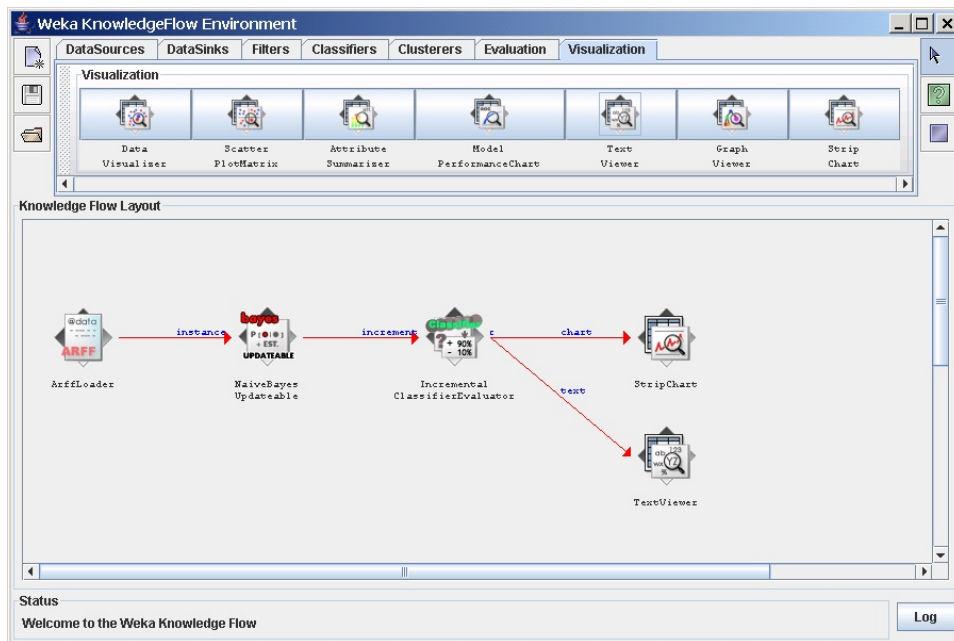


Abbildung 3.2: Weka Knowledge Flow Benutzeroberfläche (aus [WEKADOC]).

Informationen zum Weka Simple Command Line Interface finden sich unter [WEKADOC].

**Java Programmier-Schnittstelle** Weka wurde in Java implementiert. Über eine Programmier-Schnittstelle lassen sich alle Funktionalitäten von Weka in eigene Anwendungen integrieren. Außerdem ist es möglich, neue (eigene) Algorithmen zu implementieren und nahtlos in das Framework zu integrieren. Die Dokumentation zur Programmier-Schnittstelle befindet sich unter [WEKA-API].

Anhand dieser Auflistung wurden die Eigenschaften der verschiedenen Benutzer-Schnittstellen kurz vorgestellt. Im nächsten Kapitel wird unter anderem erläutert, welche dieser Benutzer-Schnittstellen für die beiden Anwendungsbeispiele am besten geeignet sind.

# Kapitel 4

## Fallstudie

In diesem Kapitel wird nun anhand der in Kapitel 1.1 und 1.2 vorgestellten Anwendungsbeispiele der Einsatz von Weka erläutert. Dabei wird aufgezeigt, wie die beiden Problemstellungen mit Hilfe von Weka umgesetzt werden können und welche Benutzer-Schnittstellen zu diesem Zweck am besten geeignet sind (siehe Kapitel 3). Jedes der Anwendungsbeispiele wird dabei in die drei Schritte “Vorverarbeitung”, “Attributanalyse” und “Anwendung der Machine-Learning Algorithmen” untergliedert (siehe Kapitel 2).

### 4.1 Anwendungsfall aus der Landwirtschaft

Das Landwirtschaftsbeispiel ist ein sehr praxisnahes Beispiel. Zu Beginn standen die Machine-Learning Experten vor dem Problem, dass sie sehr wenig über die zur Verfügung gestellten Daten wussten. Im Folgenden sind die drei wichtigsten Attribute aus dem Landwirtschafts-Merkmalssatz erläutert, wobei die ersten beiden von der *Livestock Corporation* berechnet werden:

**Production Index** Der *Production Index* gibt den Wert einer Kuh als Milchkuh an. In die Berechnung dieses Wertes fließen die produzierte Milchmenge und weitere Qualitätsmerkmale mit ein (z.B. der Eiweißgehalt). Der *Production Index* ist ein absoluter Index. Das bedeutet, dass jedem Tier eine absolute Leistungszahl zugeordnet ist.

**Breeding Index** Der *Breeding Index* gibt den Wert einer Kuh als Zuchttier an. Für die Berechnung werden beispielsweise die Anzahl der Nachkommen und deren Wert als Milchkühe miteinbezogen. Ebenso wie der *Production Index* ist der *Breeding Index* ein absoluter Index.

**Fate Code** Der *Fate Code* gibt die Ursache des Ablebens einer Kuh an. Er kann einen der Werte “sold”, “dead”, “lost” oder “unknown” annehmen. “Sold” bedeutet, dass die Kuh verkauft wurde. “Dead” besagt, dass die Kuh

geschlachtet wurde. Stirbt eine Kuh an einer Krankheit oder auf Grund einer anderen unbeabsichtigten Ursache, so ist ihr *Fate Code* "lost". Steht die Todesursache noch nicht fest, so hat ihr *Fate Code* den Wert "unknown", d.h. die Kuh lebt noch. Der *Fate Code* wird von den Bauern selbst festgelegt. Der *Breeding Index* und der *Production Index* sind aber nicht alleine ausschlaggebend für die Entscheidung, ob eine Kuh geschlachtet wird. Es ist beispielsweise auch wichtig, ob sich eine Kuh aggressiv gegenüber anderen Kühen verhält oder wie oft eine Kuh erkrankt. Ziel ist es, die Kühe nach diesen Kriterien zum Schlachten bzw. Weiterverkauf auszuwählen und so die durchschnittliche Leistung der eigenen Herde zu steigern.

Insgesamt umfassten die von der *Livestock Corporation* zur Verfügung gestellten Daten 19.000 Datensätze mit 705 Attributen. Um mehr über die Daten zu erfahren ist der Explorer die am besten geeignete Benutzeroberfläche von Weka.

#### 4.1.1 Vorverarbeitung

Die Vorverarbeitung wird an dieser Stelle in das Importieren, Visualisieren und Filtern von Daten untergliedert.

##### Importieren der Daten

Aus [GAR95] geht hervor, dass die Rohdaten in einer Datenbank abgelegt waren. Weka unterstützt den Import von Daten aus einer Datenbank. In Abbildung 4.1 ist das entsprechende Dialogfenster des Explorers zu sehen. Erwartet wird die Eingabe der Adresse der Datenbank, eine SQL-Anfrage, die die gewünschten Daten auswählt, und die Zugangsdaten. Um eine alternative Datenbankschnittstelle auszuwählen (z.B. JDBC, ODBC), kann es nötig sein weitere Einstellungen vorzunehmen (siehe [WEKADOC]).

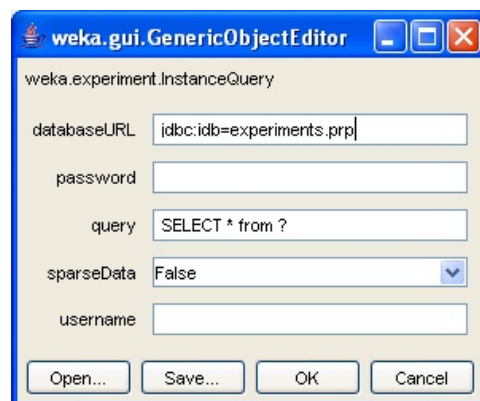


Abbildung 4.1: Dialog für den Import von Daten aus einer Datenbank.

Als nächstes sollte das Klassenattribut festgelegt werden. Hierzu wählt man im Reiter “Preprocess” des Explorers rechts unten das gewünschte Attribut aus (siehe Abbildung 3.1). Beim Landwirtschaftsbeispiel ist dies das Attribut *Fate Code* (siehe Abschnitt 4.1). Über den Knopf “Edit...” können die Werte der Datensätze, ähnlich wie in einem Tabellenkalkulationsprogramm, angesehen und bearbeitet werden. Die importierten Daten lassen sich schließlich über den Knopf “Save...” in eine Datei schreiben.

### Visualisieren der Daten

Im Reiter “Preprocess” kann man sich zu jedem Attribut folgende Werte anzeigen lassen:

- Name und Typ eines Attributs
- Anzahl der Datensätze bei denen der Wert für das ausgewählte Attribut fehlt
- Anzahl der unterschiedlichen Werte eines Attributs
- Anzahl der einmalig vorkommenden Werte eines Attributs
- Minimum, Maximum und Durchschnitt der Werte eines Attributs

Hinter dem Reiter “Visualize” verbergen sich noch weitere Möglichkeiten um verschiedene Plots der Daten zu erstellen (siehe [WEKADOC]).

Die Visualisierung der Daten eignet sich sehr gut um die Eigenschaften der Rohdaten zu untersuchen. Beispielsweise wurde in [GAR95] berichtet, dass das *Small Disjunct Problem* relativ häufig auftritt. Dies bedeutet, dass die einzelnen Klassen verschieden stark in den Daten repräsentiert sind und angeglichen werden müssen. Mit Hilfe der Visualisierungsfunktionen von Weka ist dieses Problem sehr leicht festzustellen. Im nächsten Abschnitt wird erläutert, wie man das *Small Disjunct Problem* mit Hilfe von Filtern lösen kann.

### Filtern der Daten

Weka bietet die Möglichkeit, die Eingabedaten zu filtern und bestimmte Datensätze und Attribute auszusortieren. Im Fall des Landwirtschaftsbeispiels wurden beispielsweise alle Instanzen von Kühen, bei denen das Attribut *Fate Code* den Wert “lost” hatte, aussortiert. Um diese Instanzen auszusortieren kann der Filter “weka.filters.unsupervised.instance.RemoveWithValues” eingesetzt werden.

Um das im letzten Beispiel genannte *Small Disjunct Problem* zu verhindern eignet sich der Filter “weka.filters.supervised.instance.SpreadSubsample”.

Über ihn lässt sich die absolute Anzahl von Instanzen in einer Klasse beschränken. Es kann auch ein maximaler relativer Größenunterschied zwischen

der kleinsten und der größten Klasse angegeben werden (z.B. “Anzahl der Instanzen in der größten Klasse ist maximal 1,5mal so groß wie in der kleinsten Klasse”).

Weka bietet noch wesentlich mehr Filter (z.B. für das Konvertieren von nominalen zu numerischen Attributen). Das Ergebnis der Vorverarbeitung sind saubere und konsistente Daten. Als nächstes müssen die relevanten Merkmale ausgewählt werden.

### 4.1.2 Attributanalyse

In [GAR95] wurde erwähnt, dass für die Attributauswahl die Klassifizierer *1R* und *C4.5* mit *Forward Subset Selection* zum Einsatz kamen. Im Weka Explorer findet man unter dem Reiter “Attribute Selection” alle Optionen zur Attributauswahl. Weka unterscheidet grundsätzlich in einen “Attribute Evaluator” und einen “Selection Algorithm”. Der “Attribute Evaluator” weist jedem Attribut einen Wert zu, der dessen Vorhersagekraft widerspiegelt. Der “Selection Algorithm” wählt aus den bewerteten Attributen die beste Unter-  
menge aus.

Der *1R* Algorithmus ist in der Klasse “weka.attributeSelection.OneRAttributeEval” implementiert. Möchte man den *C4.5* Algorithmus einsetzen, so kann man dies über die Klasse “weka.attributeSelection.ClassifierSubsetEval”. Als Klassifikationsalgorithmus wählt man in den Optionen schließlich “J48” (Weka Implementierung von *C4.5*).

Für *Forward Subset Selection* steht die Klasse “weka.attributeSelection.BestFirst” oder auch “weka.attributeSelection.GreedyStepwise” zur Auswahl. Darüber hinaus existieren noch Implementierungen für *Genetische Algorithmen*, *Zufallssuche*, uvm. Sind die relevanten Merkmale ausgewählt, können Machine-Learning Algorithmen auf sie angewandt werden.

### 4.1.3 Anwendung der Machine-Learning Algorithmen

Wie in [GAR95] erwähnt wurde, kam beim Landwirtschaftsanwendungsfall der *C4.5* Algorithmus zum Einsatz. Alle Einstellungen zu den Klassifikationsalgorithmen findet man unter dem Reiter “Classify” im Weka Explorer. Die Implementierung des *C4.5* Algorithmus heißt unter Weka “J48”. Über den Button “Choose” lässt sich dieser auswählen. Im Optionen-Dialog lassen sich alle Parameter des Algorithmus setzen.

Um einen erzeugten Klassifizierer zu testen bietet Weka die Optionen die Trainingsdaten als Testdaten zu nutzen, separate Testdaten zu verwenden, *Kreuzvalidierung* anzuwenden oder auch die Trainingsdaten prozentual in Test- und Trainingsdaten zu teilen. Über den Knopf “Start” lässt sich der Klassifizierer mit den eingestellten Parametern trainieren und testen. Das erstellte Ergebnisprotokoll wird in einem Fenster ausgegeben.

Es kann sich auch lohnen, weitere Machine-Learning Algorithmen auf ihre

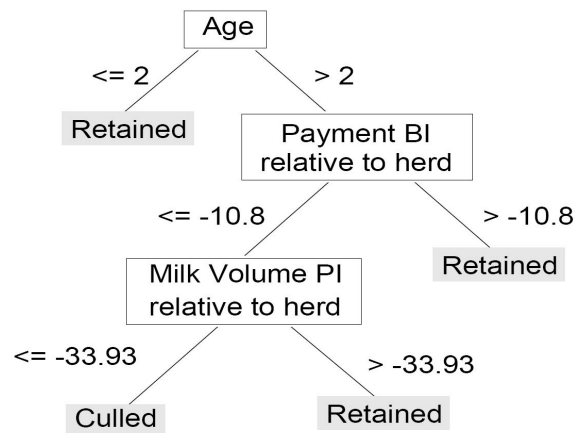


Abbildung 4.2: Klassifikationsbaum für das Landwirtschaftsbeispiel (vgl. [GAR95, S. 20]). Alle drei Attribute sind abgeleitete Attribute. Aus dem Geburtsdatum einer Kuh wurde das Alter abgeleitet und der *Breeding Index* bzw. der *Production Index* wurden relativ zur Herde berechnet.

Vorhersagekraft zu testen. Die gewonnenen Erkenntnisse lassen sich dazu nutzen, die Optionen und Parameter der eingesetzten Algorithmen noch weiter zu verbessern.

In Abbildung 4.2 ist der für das Landwirtschaftsbeispiel erzeugte Klassifikationsbaum aus [GAR95] zu sehen. Die wichtigsten Erkenntnisse die bei der Untersuchung des Problems hinsichtlich der Machine-Learning Parameter gewonnen wurden sind:

- Der Klassifikationsbaum besteht komplett aus abgeleiteten Attributen. Durch die Zusammenarbeit mit den fachlichen Experten war es möglich, “bessere” bzw. konkretere Attribute auf Basis der existierenden Attribute einzuführen, die wesentlich aussagekräftiger sind.
- Es wurden mehrere Iterationen benötigt, um die Parameter der Machine-Learning Algorithmen zu optimieren. Man hat die gewonnenen Informationen eines Durchgangs dazu genutzt, das Ergebnis noch weiter zu verbessern. Das Resultat ist ein sehr kompakter Klassifikationsbaum.

In diesem Abschnitt wurde erläutert, wie die in [GAR95] entnommene Problemstellung mit Weka umgesetzt werden kann. Für das Analysieren der Daten und das Einstellen der Algorithmen ist der Weka Explorer am besten geeignet. Durch iteratives Vorgehen, wie in Kapitel 2 beschrieben, konnte das Problem gelöst werden. Weka bietet für alle in der Literatur erwähnten Probleme entsprechende Lösungen. Im nächsten Abschnitt wird nun die Fallstudie zum *Music Information Retrieval* Beispiel vorgestellt.

## 4.2 Music Information Retrieval

Im Gegensatz zum Landwirtschaftsbeispiel ist der *Music Information Retrieval* Anwendungsfall eher theoretisch (siehe Kapitel 1.2). Hauptaufgabe war es, mit Hilfe von zwölf Algorithmen und vier Merkmalssätzen sechs verschiedene Wahrnehmungskategorien vorherzusagen (siehe Abbildung 4.3). Insgesamt ergaben sich 240 zu testende Kombinationen.

Categorization	Classes (# of songs in class)
mood	happy (29%), neutral (50%), sad (21%)
perceived tempo	very slow (4%), slow (20%), medium (43%), fast (24%), very fast (5%), varying (4%)
complexity	low (18%), medium (56%), high (7%)
emotion	soft (29%), neutral (44%), aggressive (26%)
focus	vocals (6%), both (69%), instruments (26%)
genre	blues (1%), classical (5%), electronica (13%), folk (2%), jazz (1%), new age (5%), noise (0.1%), rock (60%), world (10%)

Abbildung 4.3: Diese Abbildung zeigt die sechs Wahrnehmungskategorien mit den jeweiligen Klassen (vgl. [POH05]). Die Prozentzahlen geben den jeweiligen Anteil der Lieder an, die der entsprechenden Klasse zugeordnet wurden. Die Zuordnung erfolgte durch eine Person.

Die der Evaluation zugrundeliegenden Merkmalssätze wurden so gewählt, dass sie den üblicherweise in der *Music Information Retrieval* Literatur gewählten Merkmalssätzen entsprechen. Die einzelnen Merkmalssätze werden an dieser Stelle nur sehr knapp erläutert. Weiterführende Informationen findet man unter den jeweiligen Literaturangaben.

**Merkmalssatz 1** Der erste Merkmalssatz besteht aus insgesamt 30 Attributen und enthält Kennwerte zur *Klangfarbe*, dem *Beat*- und dem *Pitch-Histogramm* eines Liedes ([POH05], [TZA05]).

**Merkmalssatz 2** Der zweite Merkmalssatz besteht aus 20 Kennwerten, deren Definition dem MPEG 7 Format entnommen wurde [POH05]. MPEG 7 beschreibt Metadaten für Audio- und Videodaten. Beispiele für diese Attribute sind Informationen über *Audio Harmonicity* und *Audio Power*.

**Merkmalssatz 3** Der dritte Merkmalssatz besteht aus allen Attributen des ersten und zweiten Merkmalsatzes. Zusätzlich wurden noch einige weitere Merkmale, wie zum Beispiel Informationen über *Spectral Power* und die *Bandbreite*, hinzugefügt. Insgesamt hatte dieser Merkmalssatz 146 Attribute [POH05].

**Merkmalssatz 4** Der vierte Merkmalssatz wurde von Sony entwickelt und enthält hauptsächlich Informationen über die *Klangfarbe* eines Liedes ([POH05], [AUC02]). Dieser Merkmalssatz wurde daraufhin optimiert, die Ähnlichkeit von Liedern festzustellen.

Die Datenbasis für das Experiment bestand aus insgesamt 834 Liedern im MP3-Format. Jedes Lied wurde von einer Person in jeder der sechs Wahrnehmungskategorien einer Klasse zugewiesen (z.B. Klasse *slow* in der Wahrnehmungskategorie *perceived tempo*; siehe Abbildung 4.3).

Die folgenden Abschnitte stellen die Unterstützung von Weka für die “Vorverarbeitung”, “Attributanalyse” und “Anwendung der Machine-Learning Algorithmen” im Rahmen des *Music Information Retrieval* Beispiels vor.

#### 4.2.1 Vorverarbeitung

Im Gegensatz zum Landwirtschaftsbeispiel wurden die Quelldaten nicht “geliefert”, sondern mussten erst aus den Audiodaten extrahiert werden. Da es sich bei den Daten um “künstlich” generierte bzw. extrahierte Daten handelt, ist mit wesentlich weniger Anomalien zu rechnen. In [POH05] wurden keine genauen Angaben darüber gemacht, aus welcher Quelle (Datei, Datenbank, ...) die Rohdaten importiert wurden. Wie bereits erwähnt, liefert Weka aber sehr viele Möglichkeiten, um Daten zu importieren. Mit Hilfe des Weka Explorers lassen sich die Rohdaten am besten analysieren.

#### 4.2.2 Attributanalyse

Welche Algorithmen beim Auswählen der signifikanten Attribute zum Einsatz kamen, wurde im Paper nicht erwähnt. Wie bereits erläutert bietet Weka hierfür aber eine breite Palette an Algorithmen und Parametern zur Auswahl. Für die Wahl der Algorithmen und Parameter eignet sich der Weka Explorer am besten. Mit ihm ist es sehr einfach möglich, die Daten zu visualisieren und die Parameter der Algorithmen zu testen.

#### 4.2.3 Anwendung der Machine-Learning Algorithmen

Folgende Machine-Learning Algorithmen kamen zum Einsatz (manche davon in verschiedenen Ausführungen; z.B. *kNN* mit unterschiedlichen *k*):

- *kNN*

- *Naive Bayes*
- *C4.5*
- *Support Vector Machine*
- *Ada Boost*
- *M5, Linear Regression*

Mit Hilfe des Weka Explorers lassen sich die optimalen Parameter für die einzelnen Algorithmen am einfachsten herausfinden. Das Experiment selbst kann anschließend mit dem Weka Experimenter ausgeführt werden. Wie bereits in Kapitel 3 erwähnt, ist es im Experimenter möglich eine Menge von Eingabedaten und eine Menge von Algorithmen anzugeben. Als Eingabedaten wählt man die extrahierten Instanzen der jeweiligen Merkmalsätze und als Algorithmen die eben genannten Machine-Learning Algorithmen. Die insgesamt 240 Testfälle sind so relativ einfach zu konfigurieren. In [POH05] wurde außerdem erwähnt, dass die Algorithmen mit 10-facher *Kreuzvalidierung* validiert wurden. Dies lässt sich über den “Experiment Type” festlegen. Mit Hilfe des “Remote Experimenter” ist es sogar möglich die große Anzahl an Experimenten auf mehreren Rechnern parallel durchzuführen.

Mit Hilfe der in [POH05] geschilderten Untersuchung wurde festgestellt, dass keine der getesteten Kombinationen befriedigende Ergebnisse liefert. Die besten Ergebnisse wurden mit Merkmalsatz 4 in Verbindung mit *kNN* erreicht. Alle anderen Kombinationen waren kaum besser als die *Base Line* (der Fehler, wenn immer in die “größte” Klasse klassifiziert wird). Es wurde die Vermutung geäußert, dass die eingesetzten Merkmalsätze nicht für die untersuchte Problemstellung geeignet sind. Außerdem wird vermutet, dass es eine “unsichtbare” Obergrenze für nur auf den Audiodaten basierte Analysen gibt. Eine mögliche Verbesserung wäre es, externe Daten, wie den Liedtext oder ähnliches, hinzuzunehmen.

In diesem Kapitel wurde anhand von zwei Anwendungsbeispielen erläutert, wie sich die in [POH05] genannten Verfahren und Arbeitsschritte mit Weka umsetzen lassen. Für das Analysieren der Daten eignet sich der Weka Explorer am besten, für das durchführen der Experimente der Weka Experimenter. Für alle in der Literatur erwähnten Probleme bietet Weka eine entsprechende Lösung.

## Kapitel 5

# Zusammenfassung

In dieser Arbeit wurde anhand zweier Anwendungsbeispiele erläutert, wie Weka zur Umsetzung von Machine-Learning Projekten eingesetzt werden kann. Dazu wurden zuerst in Kapitel 1.1 und 1.2 die beiden Beispielszenarien vorgestellt. Im Anschluß daran wurde in Kapitel 2 das Weka zugrundeliegende Prozessmodell erläutert. In Kapitel 3 wurde die Weka Arbeitsumgebung selbst kurz vorgestellt. Abschließend wurde erläutert wie Weka genutzt werden kann, um die beiden Anwendungsbeispiele zu realisieren.

Weka ist eine ausgereifte Machine-Learning Arbeitsumgebung, die durch ihren großen Funktionsumfang und die einfache Bedienung schnell zu Ergebnissen führt. Findet man einen speziellen Algorithmus nicht im Funktionsumfang von Weka, so lässt sich Weka über die Programmier-Schnittstelle erweitern. Über die Programmier-Schnittstelle ist es außerdem möglich, die Funktionalität von Weka in eigene Anwendungen zu integrieren. Neben der Online-Dokumentation [WEKADOC] existiert außerdem noch ein eigenes Buch [WIT05], das den Umgang mit der Workbench erläutert. Zum Experimentieren und schnellen Umsetzen von Problemstellungen ist Weka bestens geeignet.

# Literaturverzeichnis

- [ARFF] <http://www.cs.waikato.ac.nz/~ml/weka/arff.html>
- [AUC02] Jean-Julien Aucouturier, Francois Pachet. *Music Similarity Measure: What's the Use?* Proc. of the 3rd International Symposium on Music Information Retrieval, 2002.
- [GAR95] S.R. Garner, S.J. Cunningham, G. Holmes, C.G. Nevill-Manning, and Witten I.H. *Applying a machine learning workbench: Experience with agricultural databases*. Proc Machine Learning in Practice Workshop, Machine Learning Conference, pages 14-21, Tahoe City, CA, USA, 1995.
- [GRO99] Heinz Lothar Grob, Frank Bensberg. *Das Data-Mining-Konzept*. Arbeitsbericht Nr. 8, 1999.  
<http://www.wi.uni-muenster.de/aw/download/publikationen/CGC8.pdf>
- [POH05] T. Pohle, E. Pampalk, G. Widmer. *Evaluation of Frequently Used Audio Features for Classification of Music Into Perceptual Categories*. Proc. of the 4th Int. Workshop on Content-Based Multimedia Indexing, Riga, Latvia, 2005.
- [ROBOT] <http://soma.npa.uiuc.edu/~ding/robo.html>
- [SPAM] <http://spamassassin.apache.org/>
- [TZA01] Georg Tzanetakis et al. *Automatic Musical Genre Classification of Audio Signals*. Proc. International Symposium on Music Information Retrieval (ISMIR), pages 205–210, Bloomington, IN, USA, October 2001.
- [TZA05] Georg Tzanetakis. *Tempo Extraction using Beat Histograms*.  
<http://www.music-ir.org/evaluation/mirex-results/articles/tempo-tzanetakis.pdf>
- [WEKA-API] <http://weka.sourceforge.net/doc.dev/>
- [WEKA-DOC] <http://www.cs.waikato.ac.nz/~ml/weka/index.html>
- [WIT05] Ian H. Witten, Eiben Frank. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, 2005, Second Edition.