Maschinelles Lernen

PD.Dr. Erica Melis

melis@dfki.de

DFKI Saarbrücken und FB14 Informatik, Universität des Saarlandes
http://www.ags.uni-sb.de/˜melis/

home page für die Vorlesung:
http://www.ags.uni-sb.de/˜melis/Vorlesung/ml01.html

Ziel der Vorlesung

- Verschiedene Ansätze des Maschinellen Lernens
- einfache Algorithmen
- wann sind welche Techniken geeignet
- theoretische Resultate, die Lernaufgaben betreffen
- praktische Anwendungen

Gliederung der Vorlesung

- Einleitung
- Lernen von Entscheidungsbäumen
- Overfitting
- künstliche neuronale Netze (backpropagation)
- Bayessches Lernen (bedingte Wahrscheinlichkeiten)
- (Regellernen)
- Instanzen-basiertes Lernen
- Analytisches Lernen (EBL)
- hybrides Lernen
- Reinforcement Lernen
- (Theorie (Abschätzungen, PACL))

- Diagnosen aufgrund EEG Daten
- Projektberichte

Projekte

1. Erarbeitung eines Vorschlags für ein Lernprojekt:
   - Problembeschreibung (warum wichtig, Datenmengen?...)
   - Forschungsziele
   - Projektplan (welche Experimente, Literatur, Expertenbefragung)
   - Individuelle Aufgaben
2. Lernen einer Diagnose aus realen medizinische Daten
   - Entscheidungsbäume
   - künstliche neuronale Netze
   - nearest neighbor

Geschichtliches: A.L. Samuel's Checker

Lernen Dame (checkers) zu spielen 1959, 1963

ZUERST: auswendig lernen = effizientes Speichern + Generalisieren (Symmetrie, Rotation)

Evaluation einer Spielsituation:

- matt: kein Zug mehr möglich
- Steinvorteil: numerisch berechnet mit Dame: Stein = 3 : 2
- Evaluationsfunktion: $\sum_i \mu_i e_i$ wo μ_i Gewicht und e_i Eigenschaft$_i$Wert:
  - ADV: wird um eins erhöht für jeden passiven Stein in der 5. und 6. Reihe und um eins erniedrigt für jeden passiven Stein in der 3. und 4. Reihe
  - CENT: wird um eins erhöht für jeden passiven Stein auf den Feldern 11,12,15,16,20,21,24,25
  - DIAG: wird um eins erhöht für jeden passiven Stein, der auf den Diagonalen zur Doppelecke steht
  - ...

Lernen der Gewichte

- Buchspiele, Anpassung mit $\frac{\#badmoves - \#goodmoves}{\#badmoves + \#goodmoves}$
- Damemeister vs. Programm
- Δ-Adjustierung (Höchstwert vs. aktuellen Wert): positive$\downarrow$, negative $\uparrow$

Software Decision Tree Learning ID3

http://www.cs.cmu.edu/afs/cs/project/theo-11/www/decision-trees.lisp

This file contains a Lisp implementation of the ID3 program for learning decision trees. It also contains the set of training examples 'play tennis'.

The code is simple enough that you should be able to inspect and modify it for your own use.

The six key functions are:

1. ID3(examples target.attribute attributes) learns a decision tree, using the algorithm from Table 3.1 in the book
2. Classify(instance tree) Given a new instance and a previously learned tree, returns the classification of that instance produced by the tree.
3. Print.tree(tree) Prints a tree in human readable form
4. Print.entity(instance) Prints an instance (e.g., one of your training examples)
5. Get.value(attribute instance) Get the value of some ATTRIBUTE of INSTANCE
6. Put.value(attribute instance value) Assign VALUE to ATTRIBUTE of INSTANCE

Software: Decision Tree Learning C4.5

- http://lestat.dfki.uni-sb.de/public_html/MLC/anwendung.html
- oder URL: ftp://ftp.cs.su.oz.au/pub/ml/
- Description: the "classic" decision-tree tool, developed by J. R. Quinlan.
- Discovery tasks: Classification
- Platform(s): Unix
- Status: Shareware
- Comments: C4.5 also includes a module for converting a decision tree to a set of rules. Patches (latest as of June 1996 is c4.8) and papers related to C4.5 are available at ftp://ftp.cs.su.oz.au/pub/ml/. There is also a more recent commercial version, called C5.0, available both for Unix and Windows.
- Contact: Morgan Kaufmann publishers. The code, including source C-code, comes with the book, C4.5 – Programs for Machine Learning, J. R. Quinlan, Morgan Kaufmann, 1993.

Software: Neuronale Netze

Neural Networks for Face Recognition

Backpropagation ALgorithmus:

http://www.cs.cmu.edu/afs/cs.cmu.edu/user/mitchell/ftp/faces.html

Software: Naive Bayesian (Text) Klassification

http://www.cs.cmu.edu/afs/cs/project/theo-11/www/naive-bayes.html

- The code is based on the Rainbow/Libbow software package developed by Andrew McCallum. It includes efficient code for indexing text documents along with code implementing the Naive Bayes learning algorithm. This code may be used as both a building block for creating other programs, or as a stand-alone learning/classification system.
- A tarred and gzipped subset of the Newsgroup data which contains 100 randomly selected messages from each newsgroup. This is a useful dataset for learning to use Rainbow.

Lernen mit Bayesschen Netzen: Netica

http://bayes.stat.washington.edu/almond/belief.html#Netica

(oder bei Th.Weyhrat, Th.Weis)

- Features: Probabilities, Utilities, Influence Diagrams, API.
- Lacks: Belief Functions, Parameter Uncertainty.
- Platforms supported: WindowsNT, Windows95 (summer 96), Unix
- Restrictions: License Required

Markus Spies (1993): Unsicheres Wissen. Wahrscheinlichkeit, Fuzzy-Logik, neuronale Netze und menschliches Denken. Heidelberg: Spektrum Akademischer Verlag.

Software: Lernen mit Bayesschen Netzen

Software: BKD: Bayesian Knowledge Discoverer, Version 1.0 (Beta)

- URL: http://kmi.open.ac.uk/projects/bkd/
- Description: BKD is a program designed to extract Bayesian Belief Networks (BBNs) from (possibly incomplete) databases. It is based on a new estimation method called Bound and Collapse and its extensions to model selection.
- Discovery tasks: Dependency analysis, belief networks
- Platform(s): Version 1.0 (Beta) for MS Windows 95/NT, MACINTOSH: Power Macintosh (System 7), UNIX: Sun Sparc (Solaris 2.4 and 2.5.1)
- Status: Research Prototype
- BKD Version 1.0 (Beta) for MS Windows 95/NT requires MS Windows 95/NT, 32 MB RAM (64 MB preferred), and 30 MB of free diskspace.
- Distribution: A copy of BKD Version 1.0 (Beta) for MS Windows 95/NT can be downloaded from the WWW site of the Bayesian Knowledge Discovery Project, The Open University at http://kmi.open.ac.uk/projects/bkd