Vorlesung "Computerlinguistische Techniken"

8. Übung (19.01.2016)

Wintersemester 2015/16 – Prof. Dr. Alexander Koller

In dieser Übung können Sie alle Programme und Programmiersprachen verwenden, die Ihnen nützlich erscheinen.

Sie brauchen außerdem das Weka-Tool, das Sie auf `http://www.cs.waikato.ac.nz/~ml/weka` herunterladen können. Weka ist ein Java-Programm, mit dem Sie Datensätze explorieren sowie eine Reihe verschiedener Klassifikations- und Regressionsalgorithmen anwenden können.

1 Klassifikation

a) Beschaffen Sie sich den Datensatz "Tic-Tac-Toe Endgame" aus dem UCI Machine Learning Repository (`http://archive.ics.uci.edu/ml`) und lesen Sie die Dokumentation. Die Daten haben einen Dateinamen der Form `*.data`; die einzelnen Spalten sind in der Datei `*.names` erklärt.

   Konvertieren Sie die Daten in ein Format, das Weka einlesen kann (per Programm oder von Hand). Am einfachsten ist vermutlich das CSV-Format:

   - Jede Zeile außer der ersten steht für eine Instanz. Jede Spalte außer der letzten steht für ein Feature.
   - In jeder Zeile stehen die Werte der einzelnen Features, durch Kommas getrennt. In der letzten Spalte, auch durch ein Komma abgetrennt, steht die Klasse der Instanz.
   - In der ersten Zeile stehen die Namen der einzelnen Features, durch Kommas getrennt. In der letzten Spalte, auch durch ein Komma abgetrennt, steht ein Name für die Klasse, den Sie frei erfinden können (z.B. "klasse" oder "gewinnt").

   Öffnen Sie den "Weka Explorer" und laden Sie Ihre Daten. Verwenden Sie den Knopf "Visualize All", um sich einen Überblick über die

Verteilung der Daten zu verschaffen. Haben Sie intuitiv ein Gefühl dafür, welche Features zur Vorhersage der Klasse nützlich sind (evtl. gar keine)? Begründen Sie Ihren Eindruck.

b) Verwenden Sie die folgenden Machine-Learning-Ansätze (im Weka-Tab "Classify"), um Klassifikatoren für Tic-Tac-Toe zu lernen. Maximum Entropy werden wir erst in der nächsten Vorlesung besprechen, aber das soll Sie nicht daran hindern, diesen Ansatz schon mal zu verwenden.

   - Memory-Based Learning (1-nearest neighbor): `lazy.IB1`
   - Naive Bayes: `bayes.NaiveBayes`
   - Maximum Entropy: `functions.SimpleLogistic`

   Geben Sie die Accuracy der einzelnen Ansätze bei 10-fold cross validation an (als Prozentzahl unter "Correctly Classified Instances").

c) Der Datensatz "Tic-Tac-Toe Endgame" beschreibt, wie der Name schon sagt, nur Endpositionen von Tic-Tac-Toe. Angenommen, wir hätten einen (viel größeren) Datensatz "Tic-Tac-Toe", aus dem wir einen Klassifikator trainieren könnten, der uns für jede beliebige Spielposition sagt, ob Spieler X gewinnt, Spieler O gewinnt, oder ein optimales Spiel unentschieden endet. Skizzieren Sie, wie ein Programm aussehen könnte, das Tic-Tac-Toe gegen einen Menschen spielt, indem es diesen Klassifikator verwendet.

d) Wiederholen Sie die Teilaufgaben (a) und (b) für den Datensatz "Mushroom" aus dem UCI-Repository. Beachten Sie, dass bei diesem Datensatz in der Datei `agaricus-lepiota.data` die Klasse in der ersten und nicht in der letzten Spalte steht.

Abgabe bis 26.01.2016, vor der Vorlesung, per Mail an `mgerdes@uni-potsdam.de`