Proseminar Maschinelles Lernen und Experimentelles Design, UdS, SS11 Hausaufgaben für den 14.7.2011

Caroline Sporleder

6. Juli 2011

Aufgabe: Die Studierenden in einem Kurs zum Thema "Maschinelles Lernen" haben als Semesterprojekt die Aufgabe bekommen, ein eigenes Lernprojekt zu implementieren. Im folgenden seht ihr die Zusammenfassung der einzelnen Projekte. Leider haben manche (alle?) Studierenden im Theorieteil nicht immer aufgepaßt, so daß sich einige Fehler eingeschlichen haben. Könnt ihr euren Kommilitonen sagen, was falsch ist und wie sie es besser machen könnten?

Anastasia Ambitioniert Anastasia hat ein Programm zur Rechtschreibprüfung entwickelt, daß Eingabeworte danach klassifiziert, ob sie korrekt oder falsch geschrieben sind. Als Daten wählt sie 1.000 Zeitungsartikel einer überregionalen Tageszeitung mit einer durchschnittlichen Artikellänge von 200 Wörtern aus. Sie annotiert jedes Wort als entweder 'korrekt' oder 'falsch'. Für ihr Experiment benutzt sie einen Decision Tree Learner. Sie kodiert eine Reihe von einfachen Attributen für die Instanzen in ihrem Datenset, z.B. ob das Wort im systemeigenen Wörterbuch vorkommt, wie häufig das Wort im Gesamtdatenset vorkommt, und ob das Wort ungewöhnliche Buchstabenkombinationen enthält. Um sicherzustellen, dass ihre Ergebnisse verlässlich sind, benutzt sie 10-fold Cross-Validation. Sie prunt ihre Entscheidungsbäume um Overfitting zu vermeiden, sorgt aber daür, dass die Bäume jeweils anhand eines Teils des Trainingssets des jeweiligen Folds geprunt werden (d.h. nicht des Testsets). Für jeden Fold berechnet sie die Accuracy auf dem Testset und bildet dann den Durchschnitt über alle Folds. Als Baseline implementiert sie einen einfachen Klassifikator, der zufällig die Ausgabeklasse (d.h. 'korrekt' oder 'falsch') auswählt, jeweils mit einer Wahrscheinlichkeit von 0.5. Diesen Klassifikator testet sie ebenfalls auf den selben Cross-Validation Folds wie ihren Lerner. Sie stellt fest, daß ihr Entscheidungsbaumlerner deutlich bessere Ergebnisse liefert als der Baseline-Klassifikator. Zur Sicherheit wendet sie einen

Signifikanztest an, der anzeigt, dass der Unterschied zwischen den beiden Klassifikatoren statistisch hoch signifikant ist. Hat sie alles richtig gemacht?

Diesbert Düsentrieb Diesbert hat ein System zur Spam-Erkennung entwickelt. Er hat ein annotiertes Datenset gefunden, das aus 1,000 Emails besteht. 400 davon sind als 'Spam' klassifiziert, 600 als 'Nicht Spam'. Um einen möglichst guten Spamklassifikator zu entwickeln schaut er sich zuerst die Emails in dem Datenset genau an. Er plant dafür mehrere Wochen ein. Während er die Emails durcharbeitet, macht er sich detaillierte Notizen über mögliche Attribute für seinen Spamklassifikator, d.h. er schreibt auf, welche Eigenschaften der Mails besonders gut geeignet erscheinen, um Spam und Nicht-Spam zu unterscheiden. Als er endlich mit der Datensichtung fertig ist, implementiert er seinen Klassifikator und kodiert die Attribute, die ihm vielversprechend erscheinen. Dann teilt er das Datenset in 80% Trainings- und 20% Testdaten. Er sorgt dafür, daß die Verteilung von Spam vs. Nicht-Spam in beiden Datensets gleich ist. Er verwendet einen Decision Tree Learner mit Pruning, stellt aber wie Anastasia sicher, daß nur das Trainingsset verwendet wird, um zu entscheiden welche Knoten geprunt werden. Er bekommt sehr gute Ergebnisse (Accuracy). Da er ein bereits existierendes Datenset verwendet, kann er zum Vergleich auf frühere Ergebnisse für dieses Datenset zurückgreifen. Er führt einen Signifikanztest durch und stellt fest, daß die Accuracy seines Systems signifikant höher ist als die des früheren Systems. Hat er alles richtig gemacht?

Clemens Clever Clemens hat ein neuartiges Information Retrieval (IR) System entwickelt, das auf Memory Based Learning basiert. Er testet sein System auf einem großen Datenset seines Instituts, das mehr als 10 Millionen Dokumente enthählt. Das Ziel des Systems ist es, zu einer Suchanfrage eines Nutzers jeweils die passenden Dokumente aus dem Datenset auszuwählen. Clemens stellt sich 20 Beispiel-Suchanfragen zusammen. Ideallerweise möchte er sowohl die Precision als auch den Recall seines Systems bestimmen (d.h. welcher Prozentsatz der zurückgegebenen Dokumente ist relevant für die Suchanfrage und welche Prozentsatz der relevanten Dokument wird gefunden). Leider würde die Berechnung des Recalls voraussetzen, daß für jeders der über 10 Millionen Dokumente annotiert wird, ob es für eine der 20 Suchanfragen relevant ist. Dafür hat Clemens keine Zeit. Stattdessen berechnet er nur die Precision. Hierfür testet er sein System mit jeder der 20 Suchanfragen und annotiert die zurückgegebenen Ergebnisse als 'relevant' bzw. 'nicht relevant'. Um eine Baseline zu haben, sucht er nach einer Publikation, die das IR System eines großen Suchmaschinen-Anbieters beschreibt. Er freut sich, weil die dort angegebene Precision geringer ist als die seines Systems. Hat er alles richtig gemacht?