Maschinelles Lernen: Symbolische Ansätze

TECHNISCHE UNIVERSITÄT DARMSTADT

Wintersemester 2008/2009
4. Übungsblatt für den 18.11.2008

Aufgabe 1: Batch-FindG, Separate-And-Conguer und Bottom-Up Regellernen

Gegeben sei der Golf-Spiel Datensatz aus der Vorlesung.

```
@relation weather.symbolic
@attribute outlook {sunny, overcast, rainy}
@attribute temperature {hot, mild, cool}
@attribute humidity {high, normal}
@attribute windy {TRUE, FALSE}
@attribute play {yes, no}
@data
sunny,hot,high,FALSE,no
sunny,hot,high,TRUE,no
overcast,hot,high,FALSE,yes
rainy,mild,high,FALSE,yes
rainy,cool,normal,FALSE,yes
rainy,cool,normal,TRUE,no
overcast,cool,normal,TRUE,yes
sunny,mild,high,FALSE,no
sunny,cool,normal,FALSE,yes
rainy,mild,normal,FALSE,yes
sunny,mild,normal,TRUE,yes
overcast,mild,high,TRUE,yes
overcast,hot,normal,FALSE,yes
rainy,mild,high,TRUE,no
```

Die positive Klasse sei die Klasse `yes`.

a) Führen Sie eine Iteration des BATCH-FINDG Algorithmus aus der Vorlesung durch. Woran erkennen Sie, daß dieses Problem nicht mit diesem Algorithmus lösbar ist?

b) Wenden Sie den Seperate-And-Conquer-Algorithmus auf die Beispiele an. Konstruieren Sie die einzelnen Regeln mittels Top-Down Hill-Climbing (Siehe Lernen von Regel-Mengen, Folie 8 bzw. 14):

- mit dem Maß Precision
- mit dem Maß Accuracy, wobei die aktuelle Regel solange verfeinert wird, bis keine negativen Beispiele mehr abgedeckt werden. Anschließend wählen Sie aus den so entstandenen Regeln diejenige aus, die die höchste Accuracy hat.

Diskutieren Sie die Ergebnisse. Welche Regelmenge sieht am besten aus?

c) Wiederholen Sie 1b), indem sie die Rolle der Klassen vertauschen (also die positive Klasse sei jetzt `no`).

d) Eine Bottom-Up Lern-Strategie (also Specific-To-General) zur Batch-Induktion einzelner Regeln könnte so aussehen, daß ein positives Beispiel zufällig ausgewählt wird, und dann sukzessive generalisiert wird. Simulieren Sie diese Strategie an diesen Trainings-Beispielen, wobei Sie aus Gründen der Vergleichbarkeit bitte als erstes “zufällig” ausgewähltes Beispiel das fünfte Beispiel verwenden.

e) Eine alternative Strategie wäre, alle Beispiele in Regeln zu verwandeln, zwei beliebige Regeln auszuwählen, das lgg dieser Beispiele zu finden, und dann die beiden alten Regeln durch diese neue zu ersetzen. Wieso wird diese Strategie i.a. nicht funktionieren? Wie könnte man sie verbessern (z.B. durch Auswahl der Regeln, Abbruchbedingungen, etc.)?

f) Überlegen Sie sich, wie dieser Algorithmus mit numerischen bzw. hierarchischen Attributen umgehen könnte.

Aufgabe 2: Grenzen der Regellerner

Gegeben sei der folgende Datensatz.

```
@relation x
@attribute a1 {0,1}
@attribute a2 {0,1}
@attribute a3 {0,1}
@attribute a4 {0,1}
@attribute x {yes, no}
@data
1,0,0,0,yes
1,1,0,1,yes
0,0,1,1,no
1,0,0,1,no
1,1,1,0,no
0,0,1,0,yes
0,0,0,1,no
1,1,0,0,no
0,1,1,1,yes
1,0,1,0,yes
0,1,0,1,yes
0,1,1,0,no
```

a) Versuchen Sie eine möglichst einfache Regelmenge zu finden oder zu lernen, die diesen Datensatz erklärt.

b) Warum hat der Separate-and-Conguer Algorithmus (unabhängig von der eingesetzten Heuristik) Probleme beim Lernen dieses Datensatz?

Aufgabe 3: Coverage Space

a) Gegeben seien Klassifizierer, die mit der Wahrscheinlichkeit p_+ für ein Beispiel unabhängig von seinen konkreten Attribut-Werten die Klasse $+$ vorhersagt. Entsprechend wird mit der Wahrscheinlichkeit $1 - p_+$ für ein Beispiel die Klasse $-$ vorhergesagt. Wo im Coverage Space liegen diese Klassifizierer für verschiedene Wahrscheinlichkeiten von p_+ (z.B. 0,2, 0,5, 0,8).

b) Overfitting aufgrund von fehlerhaften Trainings-Beispielen äußert sich oft, indem Regeln mit geringer Coverage gelernt werden. Identifizieren Sie den für Overfitting ausschlaggebenden Bereich im Coverage Space und überlegen Sie sich die Eigenschaften der in der Vorlesung besprochenen Maße bezüglich Overfitting. Z.B. welches Maß neigt eher zu Overfitting, Precision oder Accuracy?