Data Mining und Maschinelles Lernen

TECHNISCHE UNIVERSITÄT DARMSTADT

Wintersemester 2015/2016
4. Übungsblatt

Aufgabe 3 aus vorheriger Übung

Gegeben sei ein Datensatz mit drei Attributen:

Haarfarbe: *blond*, *braun*, *schwarz*
Größe: *klein*, *groß*
Augenfarbe: *grün*, *blau*

Der Hypothesenraum besteht aus Disjunktionen (Oder-Verknüpfungen) von maximal einem Wert pro Attribut, einer speziellsten Theorie *false*, die keine Beispiele abdeckt, und einer allgemeinsten Theorie *true*, die alle Beispiele abdeckt.

Zum Beispiel deckt die Hypothese *blond* $\vee$ *blau* alle Personen ab, die entweder blond oder blauäugig sind (in der Datenmenge aus Aufgabe 3b) sind das z.B. die Beispiele 1, 3, 4).

Beachte: Hypothesen wie *blond* $\vee$ *braun*, die mehrere Werte desselben Attributs verwenden, sind nicht im Hypothesenraum.

a) Geben Sie in dieser Hypothesensprache alle minimalen Generalisierungen und Spezialisierungen der Hypothese *blond* $\vee$ *blau* an.

b) Folgende Beispiele treffen in dieser Reihenfolge ein:

1	*braun*	*groß*	*blau*	+
2	*braun*	*klein*	*grün*	−
3	*schwarz*	*klein*	*blau*	−
4	*blond*	*klein*	*grün*	+

Das erste Beispiel kodiert also eine Person, die braune Haare und blaue Augen hat und groß ist.

Führen Sie auf diesen Beispielen den Candidate-Elimination Algorithmus zur Berechnung des Version Spaces durch und geben Sie nach jedem Schritt das S-Set und das G-Set an.

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

Gegeben sei der Golf-Spiel Datensatz aus der Vorlesung.

Outlook	Temperature	Humidity	Windy	Class
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 unter Verwendung konjunktiver Regeln lösbar ist?

b) Wenden Sie den Separate-And-Conquer-Algorithmus (siehe Foliensatz *Learning Rule Sets*, Folie *Separate-and-Conquer Rule Learning*) auf die Beispiele an. Konstruieren Sie die einzelnen Regeln mittels Top-Down Hill-Climbing (siehe *Learning Individual Rules and Subgroup Discovery*, Folie *Top-Down Hill-Climbing* und *Top-Down Hill-Climbing in Coverage Space*):

   - mit dem Maß Precision
   - mit dem Maß Accuracy

   Wobei die aktuelle Regel so lange verfeinert wird, bis keine negativen Beispiele mehr abgedeckt werden. Anschließend wählen Sie aus den so entstandenen Regeln diejenige aus, die den höchsten heuristischen Wert hat. Als Tie-Breaking wählen Sie zunächst diejenige Regeln aus, die am meisten positive Beispiele abdeckt. Sollte dies nicht ausreichen, wählen Sie die zuerst gefundene Regel aus.

   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. In allen weiteren Iterationen wählen Sie bitte das erste positive Beispiel der verbleibenden Trainingsmenge. Als Abbruchkriterium gilt hier das Erreichen der generellsten Regel (die selbst nicht mehr mitbetrachtet wird).

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 der Separate-And-Conquer-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-Conquer-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?