Semestralklausur zur Vorlesung

Maschinelles Lernen: Symbolische Ansätze

Prof. J. Fürnkranz / Dr. G. Grieser
Technische Universität Darmstadt — Wintersemester 2005/06
Termin: 23. 2. 2006

Name: **Vorname:** **Matrikelnummer:**

Fachrichtung:

Punkte: (1) .... (2) .... (3) .... (4) .... (5) .... **Summe:**

Aufgabe 1 (15 Punkte)

1–a Sie haben Q-learning verwendet, um eine Funktion zu lernen, die Ihnen aus einem Labyrinth hinaushilft. Sie befinden sich im Zentrum und können nach Norden, Westen, Süden oder Osten gehen. Wie setzen Sie die gelernte Funktion ein, um die richtige Richtung auszuwählen?

Lösung: (3 Punkte)

Man wählt die Aktion $a \in \{N, S, O, W\}$ aus, für die die gelernte Funktion $Q(s_z, a)$ maximal wird, wobei s_z der Zustand ist, der dem Zentrum des Labyrinths entspricht.

1–b Welchen Algorithmus würden Sie wählen, wenn Ihre Anwendung verlangt, daß die Anzahl der *false negatives* möglichst gering gehalten werden soll?

☐ Find-S

☐ Find-G

☐ Candidate Elimination

Begründung?

Lösung: (3 Punkte)

False Negatives sind positive Beispiele, die irrtümlich als negativ klassifiziert werden. Da der Algorithmus Find-G nur dann spezialisiert, wenn unbedingt notwendig (d.h., um ein negatives Beispiel (also ein false positive) auszuschließen), hat er zu jedem Zeitpunkt die Theorie, die so viele Beispiele wie möglich positiv klassifiziert. Falls die Theorie im Suchraum ist, wird Find-G daher keine *false negatives* produzieren.

1–c Erklären Sie kurz wozu der `FilteredClassifier` in Weka nützlich ist.

Lösung: (3 Punkte)

Mit einem `FilteredClassifier` kann man aus einer Vorerarbeitungsmethode und einem Classifier einen neuen Classifier zusammensetzen. Das ist z.B. nützlich, um Vorverarbeitungsschritte innerhalb einer Cross-Validation durchführen zu können.

1–d Gegeben sei eine Datenbank über 100.000 Informatik-Fachbücher. Jedes Buch ist durch eine Menge von Stichwörtern beschrieben, insgesamt werden in der Datenbank 1000 solcher Stichwörter benutzt.

Ihre Aufgabe sei nun, ein Programm zu schreiben, das aus der Datenbank alle Bücher zum Thema *Künstliche Intelligenz* heraussucht. Als Ausgangsbasis haben Sie eine Menge von 1000 Büchern, von denen Sie wissen, ob sie zum Thema *Künstliche Intelligenz* gehören oder nicht.

Wie würden Sie herangehen?

Lösung: (4 Punkte)

Hier gibt es kein richtig oder falsch. Bewertet wird, wie sinnvoll das beschriebene Herangehen ist.

- es handelt sich ganz klar um eine Klassifikationsaufgabe
- Auffassen der Stichwörter als boolsche Attribute, d.h. jedes Buch ist durch 1000 Attribute beschrieben
- Anwendung eines Lernverfahrens auf die 1000 gelabelten Bücher zur Erzeugung eines Klassifikators (evtl. vorher Feature subset selection)
- Benutzen dieses Klassifikators zur Klassifikation der restlichen

---

1–e Nehmen Sie an, Sie haben eine Beispielmenge mit 10 Attributen, von denen 9 Attribute exakte Kopien voneinander sind.

Welchen Effekt erwarten Sie, wenn Sie Naive Bayes auf diese Daten anwenden?

Lösung: (2 Punkte)

Die mit dem neunfach kopierten Attribut assoziierter bedingte WK geht mit der Potenz (dem Gewicht) 9 in die Berechnung ein, dadurch wird es sehr großen Einfluß auf die Klassifikation haben, d.h. das andere Attribut dominieren.

Aufgabe 2 (20 Punkte)

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 b 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.

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

Lösung: (4 Punkte)

Durch Hinzufügen einer weiteren disjunktiven Terms kann man nur mehr Beispiele abdecken, daher generalisiert diese Operation. Wegstreichen eines Teilterms spezialisiert dagegen.

Minimale Generalisierungen = { *blond* $\vee$ *blau* $\vee$ *klein*, *blond* $\vee$ *blau* $\vee$ *groß*}

Minimale Spezialisierungen = { *blond*, *blau*}

2–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.

Lösung: (12 Punkte)

G_0 = { true } (0.5 Punkte)
S_0 = { false } (0.5 Punkte)

G_1 = { true }
S_1 = { *braun*, *groß*, *blau*} (das sind *drei* Hypothesen!) (2 Punkte)

G_2 = { *blond* $\vee$ *groß* $\vee$ *blau*, *schwarz* $\vee$ *groß* $\vee$ *blau*} (2 Punkte)
S_2 = { *groß*, *blau*} (die Hypothese *braun* würde nun inkonsistent sein). (1 Punkt)

G_3 = { *blond* $\vee$ *groß*} (*schwarz* $\vee$ *groß* $\vee$ *blau* wird inkonsistent und zu *groß* spezialisiert, was spezieller ist als *blond* $\vee$ *groß*) (2 Punkte)
S_3 = { *groß*} (*blau* wird nun inkonsistent) (1 Punkt)

G_4 = { *blond* $\vee$ *groß*}
S_4 = { *blond* $\vee$ *groß*} (auf *groß* $\vee$ *grün* können wir nicht generalisieren, da diese Theorie keine Spezialisierung eines Elements in G ist). (2 Punkte)

Der Version Space hat also zu einer einzigen Lösung konvergiert. (1 Punkt)

2–c Nehmen Sie an, die Beispiele aus Aufgabe b kämen in umgekehrter Reihenfolge. Was wäre dann das Resultat des Candidate-Elimination Algorithmus? Begründung?

Lösung: (4 Punkte)

Die Lösung wäre dieselbe, da der Version Space die Menge aller Theorien ist, die eine Menge von Beispielen konsistent und vollständig abdecken. Da die Menge von Beispielen dieselbe ist, muß auch der Version Space derselbe sein.
(Die Mengen in den Schritten S_1/G_1 bis S_3/G_3 werden natürlich andere sein.)

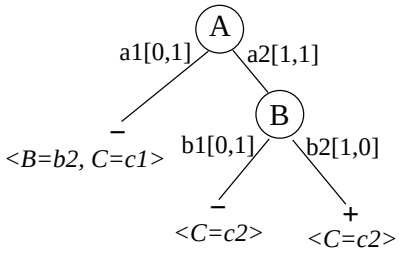
Aufgabe 3 (25 Punkte)

Diese Aufgabe bezieht sich auf ID5R.

Statt des original benutzten Maßes *Gain* benutzen wir hier eine vereinfachte Variante:

> Wähle denjenigen Test t, der folgenden Ausdruck maximiert:
> $$\sum_{v \in Werte(t)} (|S_v^+| - |S_v^-|)^2$$
> wobei S_v^+ die Menge aller Beispiele ist, für die der Test t den Wert v hat und deren Klassifikation positiv (+) sind. (S_v^- analog).
>
> Das Maß summiert also die quadrierte Differenz der Anzahl der positiven und negativen Beispiele in einer Menge.
>
> Falls mehrere Tests den gleichen Wert liefern, nimm denjenigen, der im Alphabet zuerst kommt, d.h. A vor B vor C):

Nehmen wir an, ID5R habe den folgenden Baum erzeugt:



3–a Geben Sie alle Beispiele an (d.h. lesen Sie diese aus dem Baum ab), die zur Erzeugung des Baumes geführt haben.

Lösung: (3 Punkte)

$< A = a1, B = b2, C = c1, - >$
$< A = a2, B = b1, C = c2, - >$
$< A = a2, B = b2, C = c2, + >$

Das neue Beispiel gehört hier nicht hin!

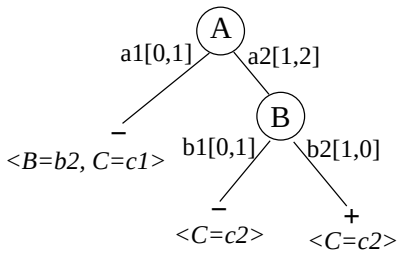
3–b Nun kommt folgendes Beispiel hinzu:

$$\langle A = a2, B = b1, C = c1; - \rangle$$

Wie lautet die neue Hypothese von ID5R? Erläutern Sie dabei Ihre Vorgehensweise, geben Sie die einzelnen Schritte an. Je detaillierter Sie Ihren Lösungsweg beschreiben, desto mehr erleichtern Sie uns die Vergabe von Punkten.

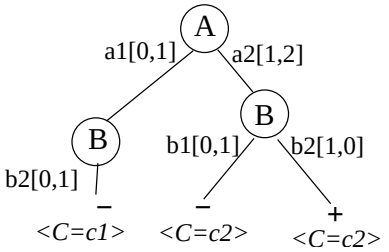
Lösung: (17 Punkte)

1. Aktualisiere Counts für die Tests im Wurzelknoten. (Da wir nicht auch noch Counts für die einzelnen Tests mitprotokollieren, heißt das hier nur, daß die Counts in den beiden ausgehenden Kanten aktualisiert werden.)

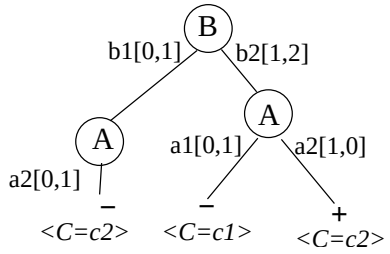


2. Bestimme, ob Test in Wurzelknoten immer noch der beste ist
   - Gain für Test A: $(0-1)^2 + (1-2)^2 = 2$
   - Gain für Test B: $(0-2)^2 + (1-1)^2 = 4$
   - Gain für Test C: $(0-2)^2 + (1-1)^2 = 4$

   → Test B ist der beste!
3. Ist er nicht → Test B in Wurzel bringen
   (a) B in linken Teilbaum etablieren
   - neuen Teilbaum mit Wurzel B wachsen lassen

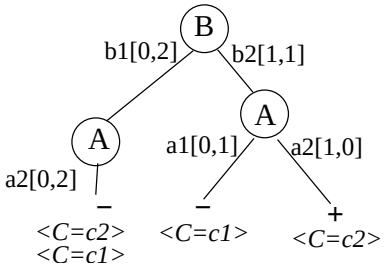


   (b) Tests A und B austauschen



4. Rekursiv die besten Tests in den Teilbäumen etablieren
   (a) linker Teilbaum: ok
   (b) rechter Teilbaum:
   - Gain für Test A: $(0-1)^2 + (1-0)^2 = 2$
   - Gain für Test C: $(0-1)^2 + (1-0)^2 = 2$

   → Test A ist der beste!
   → Baum ist nun ok
5. Baum entlang des Pfades des neuen Beispiels aktualisieren
   - Wurzel ist bereits der beste Test
   - Aktualisiere Counts an Kante von B zu A
   - Teste, ob A noch der beste Test ist → ja
   - Aktualisiere Counts an Kante von A zum Blatt



   - Speichere Beispiel im Blatt <C=c1>

3–c Eine Möglichkeit, aus unvollständigen Beispielen zu lernen ist, Wahrscheinlichkeiten für jeden möglichen Attributwert zuzuweisen und Anteile der Beispiele in die Teilbäume zu propagieren (siehe Folie 34 im Satz *Entscheidungsbäume*).

Diskutieren Sie diese Idee für ID5R. (Z.B.: Welches grundlegende Problem taucht hier auf? Wie kann man damit umgehen?)

Lösung: (5 Punkte)

Eigentlich gibt es eine ganze Reihe von Problemen:

- Ein Problem ist, daß die Beispiele nacheinander verarbeitet werden. Somit sind die Anteilsabschätzungen sehr ungenau. Wenn im Extremfall das erste Beispiel unvollständig ist, dann sind noch überhaupt keine Abschätzungen vorhanden.
- Will man den beschriebenen Ansatz seriös durchziehen, dann müßte man irgendwo im Baum auch noch eine Tabelle der relativen Attributhäufigkeiten halten und diese jedes Mal aktualisieren. Im Moment sind diese über den ganzen Baum verteilt, d.h. man muß jedes Mal den gesamten Baum durchsuchen.
- Außerdem führt potentiell eine Aktualisierung der Gewichte zu einer ständigen Umstrukturierung, und zwar nicht nur in dem durch das aktuelle Beispiel beschriebenen Pfad.
- Die Beispiele werden in den Blättern gespeichert, zu jedem Beispiel gibt es einen eindeutigen Pfad. Bei anteiligen Beispielen müßten dann potentiell mehrere Pfade aktualisiert werden, und die Beispiele müßen auch mehrfach gespeichert werden.

Wir erwarten, daß mind. eines dieser Probleme erkannt und ein vernünftiger Vorschlag gemacht wurde. Aussagen wie *Die aktuelle Formel für Gain berücksichtigt keine Anteile* oder Die Zähler sind im Moment ganzzahlig sind trivial und bringen keine Punkte.

Aufgabe 4 (22 Punkte)

Gegeben sei ein Datensatz mit 300 Beispielen, davon 2/3 positiv und 1/3 negativ.

4–a Ist die Steigung der Isometrien für Accuracy im Coverage Space für dieses Problem

$\square > 1$

$\square = 1$

$\square < 1$?

Lösung: (2 Punkte)

Die Steigung von Accuracy im Coverage Space ist immer $= 1$, da positive und negative Beispiele immer gleich gewichtet werden.

4–b Ist die Steigung der Isometrien für Accuracy im ROC Space für dieses Problem

$\square > 1$

$\square = 1$

$\square < 1$?

Lösung: (2 Punkte)

Die Steigung von Accuracy in ROC Space ist < 1, da die P-Achse gestaucht wird (um einen Faktor 2, i.e., die Steigung ist 1/2).

4–c Sie verwenden einen Entscheidungsbaum, um die Wahrscheinlichkeit für die positive Klasse zu schätzen. Sie evaluieren drei verschiedene Thresholds t (alle Beispiele mit einer geschätzten Wahrscheinlichkeit $> t$ werden als positiv, alle anderen als negativ klassifiziert) und messen folgende absolute Anzahlen von false positives und false negatives:

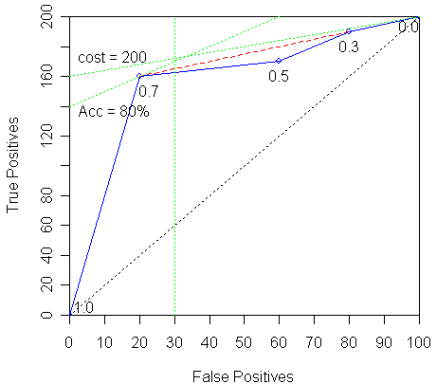
t	fn	fp
0.7	40	20
0.5	30	60
0.3	10	80

Geben Sie für jeden Threshold an, für welchen Bereich des Kostenverhältnisses $\frac{c(+|-)}{c(-|+)}$ der Threshold optimal ist.

Lösung: (8 Punkte)

200 positive Beispiele, 100 negative Beispiele

t	fn	fp	tp
0.7	40	20	160
0.5	30	60	170
0.3	10	80	190



Ein Blick auf die Kurve zeigt, daß der Threshold 0.5 in einer Konkavität liegt. Daher

- (3 Punkte) Der Threshold $t = 0.5$ ist für keinen Kosten-Bereich optimal.

Die konvexe Hülle der Kurve hat drei verschiedene Segmente:

- Die Steigung im Abschnitt von $(0, 0)$ bis $(20, 160)$ ist $160/20 = 8$.
- Die Steigung im Abschnitt von $(20, 160)$ bis $(80, 190)$ ist $30/60 = 1/2$
- Die Steigung im Abschnitt von $(80, 190)$ bis $(100, 200)$ ist $10/20 = 1/2$

Daraus folgt (5 Punkte)

- Der Threshold $t = 0.7$ ist optimal für den Bereich $1/2 \leq \frac{c(+|-)}{c(-|+)} \leq 8$ optimal.
- Der Threshold $t = 0.3$ ist nur für ein Kostenverhältnis von $1/2$ optimal.

4–d Wie hoch ist die maximale Genauigkeit (Accuracy), die Sie im Szenario von Punkt c bei einer false positive rate von maximal 30% erreichen können? Wie gehen Sie dabei vor?

Lösung: (3 Punkte)

Durch Parallel-Verschieben der Accuracy-Isometrien trifft man zuerst auf den Punkt $(20, 160)$. Die Einschränkung, daß die False Positive Rate < 0.3 sein muß, ist von diesem Punkt erfüllt.

Die Genauigkeit an diesem Punkt ist $\frac{300-(40+20)}{300} = 80\%$.

4–e Sie erfahren, daß in Ihrer Anwendung ein false positive 2 Cents kostet, und ein false negative 5 Cents kostet. Mit welchem Threshold können Sie die Kosten minimieren? Wie hoch sind die entstandenen minimalen Kosten für diese 300 Beispiele?

Lösung: (3 Punkte)

Da das Kostenverhältnis $\frac{2}{5} < 1/2$ ist, ergibt sich unmittelbar aus c daß der Punkt $(100, 200)$ optimal wäre, i.e., kein Beispiel als positiv zu klassifizieren (Threshold 1.0).

Man kann die Lösung natürlich auch durch direktes Berechnen

t	fn	fp	Kosten
1	200	0	$5 \times 200 + 0 \times 0 = 1000$
0.7	40	20	$5 \times 40 + 2 \times 20 = 220$
0.5	30	60	$5 \times 30 + 2 \times 60 = 270$
0.3	10	80	$5 \times 10 + 2 \times 80 = 210$
0.0	0	100	$0 \times 0 + 2 \times 100 = 200$

oder durch Parallel-Verschieben der Kosten-Isometrie mit Steigung 2/5 (analog zu d) finden.

Die Gesamtkosten sind 200 cents (siehe oben).

4–f Sie bekommen die Möglichkeit, zusätzlich zu den vorhandenen 300 Beispielen noch 400 selbst auszuwählen. Wie würden Sie die Auswahl treffen, damit ein Lerner, der Kosten nicht berücksichtigen kann, unter den in Punkt e angegebenen Kosten möglichst effektiv wird?

Lösung: (4 Punkte)

Man sollte versuchen, durch die Verteilung der Beispiele das Kostenverhältnis widerzuspiegeln, also positive zu negative Beispiele im Verhältnis 5:2 herzustellen, d.h. man sollte zusätzlich 300 positive und 100 negative Beispiele auswählen.

Aufgabe 5 (18 Punkte)

Sie wissen, daß die Assoziationsregel

`beatles`, `stones` → `dylan`, `cohen`

in einem Datensatz mit 1000 Einträgen über Musikpräferenzen einen Support von 0.4 und eine Konfidenz von 0.8 hat.

5–a Beantworten Sie folgende Fragen unter Angabe eines möglichst kleinen Intervalls der möglichen Werte (kann auch $[-\infty,+\infty]$ oder auch nur ein einziger Wert sein):

- Wie groß ist der Lift dieser Regel?

Lösung: (3 Punkte)

Für den Lift braucht man den Support von `dylan`, `cohen`. Der muß ≥ 0.4 und ≤ 1.0 sein.
0.8 / 0.4 = 2.0
0.8 / 1.0 = 0.8
daher lift ∈ **[0.8,2.0]**.

- Wie viele Personen mögen `beatles` und `stones`?

Lösung: (3 Punkte)

conf(A → B) = supp(A,B) / supp(A)
daher supp(A) = supp(A,B) / conf(A → B) = 0.4 / 0.8 = 0.5, i.e. **500** Personen

- Wie viele Personen mögen `stones` und `dylan`?

Lösung: (2 Punkte)

Teilmenge von `beatles`, `stones`, `dylan`, `cohen`, daher mindestens 400, also **[400,1000]**

- Wieviel Prozent der Personen, die `beatles`, `dylan`, und `stones` mögen, mögen auch `cohen`?

Lösung: (3 Punkte)

Überführen einer Bedingung vom Head in den Body kann die Konfidenz nicht senken (Folie 11), daher **[80%,100%]**.

- Wie viele Personen, die `stones` mögen, mögen auch `beatles`, `dylan`, und `cohen`?

Lösung: (3 Punkte)

Der Support einer Regel ist immer gleich, solange die gleichen Bedingungen auf Head and Body verteilt werden, daher **400** Personen.

5–b Sie wissen, daß Apriori die oben angegebenen Assoziationsregel gefunden hat.

Zusätzlich erfahren Sie noch, daß folgende Itemsets frequent waren:

{ `beatles`, `dylan`, `young` }

{ `beatles`, `young`, `stones` }

{ `young`, `dylan`, `stones` }

Geben Sie die positive und die negative Border an.

Lösung: (4 Punkte)

(2 Punkte) Positive Border = { beatles, stones, dylan, cohen }

(2 Punkte) Negative Border = { beatles, dylan, young, stones } da aufgrund der Regel auch {beatles, dylan stones} frequent sein muss.

Siehe Definitionen auf Folie 9.