

# Introduction to Data and Knowledge Engineering Sommersemester 2010



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

---

Tutorium 9: Induktives Lernen

18. Juni 2010

---

---

## Aufgabe 9.2 Finden von Literalen

---

In einem Verein soll ein neuer Vorsitzender gewählt werden. Es soll ein neues Prädikat  $\text{votes\_for}(A, B)$  gelernt werden, welches bestimmt, wann ein Vereinsmitglied A für ein anderes Vereinsmitglied B stimmt. Das Hintergrundwissen ist mithilfe der folgenden Prädikate angegeben:

- $\text{candidate}(A)$  - A ist ein Kandidat bei der Wahl zum Vorsitzenden.
- $\text{friend}(A, B)$  - A ist mit B befreundet.

Wir wollen uns in dieser Aufgabe darauf beschränken, alle Literale aufzulisten, welche zum Aufstellen des Prädikats  $\text{votes\_for}(A, B)$  überprüft werden müssten (d.h. alle Prädikate des Hintergrundwissens müssen mit allen möglichen Variablenkombinationen überprüft werden). Zur Vereinfachung verzichten wir auf das Einführen neuer Variablen in Literalen, d.h. es werden nur Variablen betrachtet, die bereits im zu lernenden Prädikat vorkommen.

---

## Aufgabe 9.3 Heuristiken

---

In Separate-and-Conquer-Algorithmen kann zur Auswahl von Literalen eine Heuristik verwendet werden, welche nach Möglichkeit einen hohen Wert liefern soll, wenn die Hinzunahme des Literals die Regel verbessert. Was genau das heißt, ist schwierig zu definieren, daher kann es keine Heuristik geben, die in jedem Fall richtig ist. Trotzdem erscheinen manche Heuristiken in der Mehrheit der Fälle „besser“ zu sein als andere.

Überlegen Sie, welche der folgenden Formeln sinnvolle Heuristiken sind und diskutieren Sie über Vor- und Nachteile der jeweiligen Formel bzw. ob Formeln gleichwertig sind.

$p$  und  $n$  beschreiben jeweils die Anzahl der positiven bzw. negativen Trainingsbeispiele, die durch diese Regel abgedeckt werden.

- $p$
- $p - n$
- $(p - n)^2$
- $(p - n)^3$
- $\frac{p}{p+n}$

---

## Aufgabe 9.4 Gain

---

Den „Gain“ einer Regel berechnet man, in dem man den Wert der gewählten Heuristik zur Auswahl von Literalen sowohl für die vorige Regel als auch für die neue Regel, die unter Hinzunahme eines Literals entsteht, berechnet und den alten Gain vom neuen subtrahiert.

Geben Sie für die Heuristiken aus der vorigen Aufgabe jeweils die zugehörigen Gain-Formeln an.

Verwenden Sie  $p$  und  $n$  für die Anzahl der abgedeckten positiven bzw. negativen Beispiele aus der neuen Regel,  $p'$  und  $n'$  für die Anzahl der positiven bzw. negativen abgedeckten Beispiele in der vorherigen Regel (vor Hinzunahme des Literals).

---

## Aufgabe 9.5 Separate-and-Conquer

---

In dieser Aufgabe betrachten wir Pferde, Esel und deren Kreuzungen. Das Hintergrundwissen ist mithilfe der folgenden Prädikate angegeben:

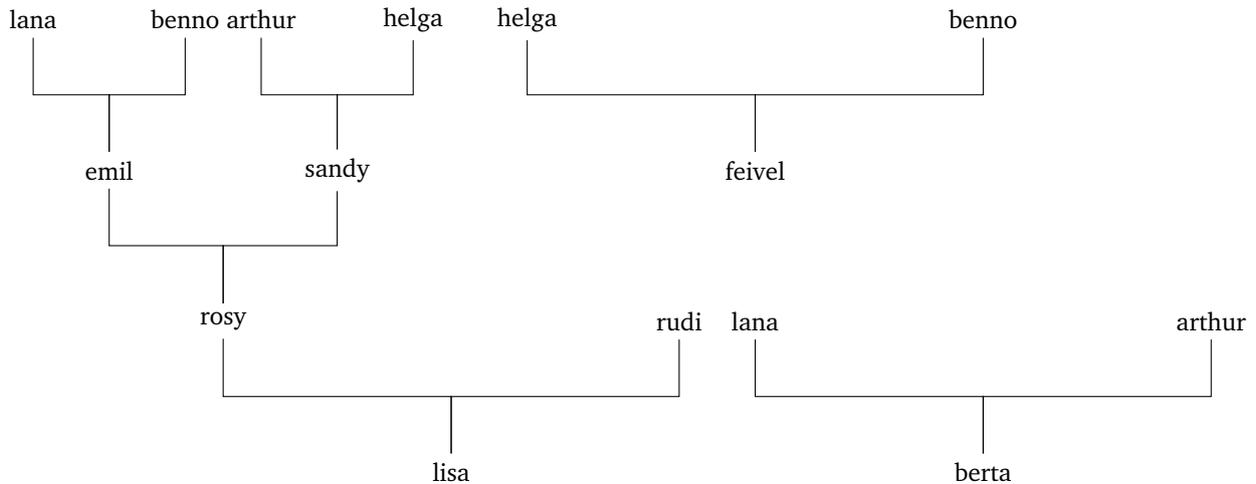
- `horse(A)` - A ist ein Pferd.
- `donkey(A)` - A ist ein Esel.
- `father(A, B)` - A ist Vater von B.
- `mother(A, B)` - A ist Mutter von B.

Bekannte Fakten:

```
horse(helga)
horse(arthur)
horse(rudi)
horse(sandy)
donkey(benno)
donkey(lana)
donkey(emil)
mother(helga, feivel)
father(benno, feivel)
mother(helga, sandy)
```

```
father(arthur, sandy)
mother(lana, berta)
father(arthur, berta)
mother(lana, emil)
father(benno, emil)
mother(sandy, rosy)
father(emil, rosy)
mother(rosy, lisa)
father(rudi, lisa)
```

Hier der Übersichtlichkeit halber die graphische Veranschaulichung der Familienverhältnisse:



Es soll ein Prädikat `mule(A)` trainiert werden, welches bestimmt, wann ein Tier A ein Maultier ist.<sup>1</sup> Hierfür sind folgende positive und negative Beispiele gegeben:

- |                             |                            |
|-----------------------------|----------------------------|
| ⊕ <code>mule(feivel)</code> | ⊖ <code>mule(sandy)</code> |
| ⊕ <code>mule(rosy)</code>   | ⊖ <code>mule(berta)</code> |
| ⊕ <code>mule(lisa)</code>   | ⊖ <code>mule(emil)</code>  |

---

<sup>1</sup> Interessantes Hintergrundwissen: Maultiere entstehen aus der Kreuzung einer Pferdestute und eines Eselhengstes. Ein Nachkommen aus dem umgekehrten Fall bezeichnet man als Maulesel. Sowohl Maulesel als auch Maultiere sind grundsätzlich nicht zu Nachkommen fähig, jedoch sind einige seltene Fälle bekannt in denen Maultierstuten geworfen haben.

Wir betrachten zur Vereinfachung nur bestimmte unten gegebene Kandidaten-Literale, um das Prädikat zu trainieren. Als Formel für den Gain verwenden wir die Differenz der Precision:

$$\frac{p}{p+n} - \frac{p'}{p'+n'}$$

$p$ ,  $n$ ,  $p'$  und  $n'$  sind analog zur vorigen Aufgabe definiert. Am Anfang ist eine Regel leer, d.h. es werden alle positiven, aber auch alle negativen Beispiele abgedeckt. Im vorliegenden Fall ist also am Anfang  $p' = 3$  und  $n' = 3$ .

**Hinweise zur Bearbeitung:**

In jeder Tabelle soll eine Regel des Prädikats gelernt werden. In jeder Spalte der Tabelle werden dabei jeweils die Werte für  $p$ ,  $n$  und Gain angegeben, unter Berücksichtigung der in vorherigen Spalten gewählten Literale (d.h. in jeder Spalte wird jedes Kandidaten-Literal zusammen mit den bereits gewählten Literalen betrachtet, zudem werden nur die positiven und negativen abgedeckten Beispiele betrachtet, welche das vorher ausgewählte Literal abdeckt). Wird schließlich ein Literal gefunden, welches keine negativen Beispiele mehr abdeckt, ist die momentane Regel fertig entwickelt. Die von dieser Regel abgedeckten positiven Beispiele werden entfernt. Sind anschließend noch positive Beispiele übrig, wird die Prozedur wiederholt.

Es soll stets das Literal mit dem absolut höchsten Gain ausgewählt werden. Wir legen als Konvention fest, dass wir im Falle von mehreren Literalen mit dem gleichen Gain das oberste auswählen. Überlegen Sie, ob die Auswahl eines der jeweils anderen Literale im vorliegenden Fall die zu lernenden Regeln wesentlich verändern würde.

Einige Kandidaten-Literale führen neue Variablen ein. Diese werden jedoch nur überprüft, falls der neu eingeführten Variable ein eindeutiger Wert zugewiesen werden kann, da ansonsten Regeln nicht sinnvoll gebildet werden könnten. Weiterhin wird natürlich ein Literal, welches bereits in die Regel aufgenommen wurde, nicht noch einmal überprüft.

1. Regel:

	p	n	Gain									
mother(B, A)												
father(A, C)												
father(C, A)												
horse(A)												
horse(B)	-	-	-									
horse(C)	-	-	-									
donkey(B)	-	-	-									
donkey(C)	-	-	-									
mule(B)	-	-	-									

2. Regel:

	p	n	Gain									
mother(B, A)												
father(A, C)												
father(C, A)												
horse(A)												
horse(B)	-	-	-									
horse(C)	-	-	-									
donkey(B)	-	-	-									
donkey(C)	-	-	-									
mule(B)	-	-	-									