Maschinelles Lernen: Symbolische Ansätze

Musterlösung für das 6. Übungsblatt

Aufgabe 1

Gegeben sei eine Beispielmenge mit folgenden Eigenschaften:

- Jedes Beispiel ist durch 10 nominale Attribute $A_1, \ldots, A_{10}$ beschrieben.
- Jedes dieser Attribute hat 2 Werte.

a) Wie viele Entscheidungsbäume müßten bei vollständiger Suche untersucht werden (es genügt eine grobe Abschätzung der Größenordnung)?
(Hinweis: Dies ist analog zu der Frage: Wie viele Entscheidungsbäume gibt es ungefähr für diese Daten?)

Lösung: Es wird in den folgenden Aufgaben jeweils eine grobe Abschätzung gegeben, die abhängig von der zur Erstellung des Baumes verwendeten Methode durchaus variieren kann.

Um die Anzahl an möglichen Bäumen abschätzen zu können, müssen wir uns überlegen, wie viele Möglichkeiten es gibt, einen vollbesetzten, balancierten Baum aufzubauen (also einen mit maximaler Anzahl an Knoten). Dazu gliedern wir die Abschätzung in 2 verschiedene Teilabschätzungen.

Als erstes wollen wir die Blätter betrachten. Da wir davon ausgehen, dass es 10 verschiedene Attribute gibt, ist die maximale Tiefe des Baumes 10. Da nun in jeder Ebene des Baumes pro Knoten jeweils 2 Unterscheidungen gemacht werden können (da jedes Attribut 2 verschiedene Werte annehmen kann), gibt es 2^{10} Blätter. Als nächstes müssen wir uns überlegen auf wie viele verschiedenen Wegen die Klassenwerte "+" und "-" in den Blättern angeordnet werden können. Da alle verschiedenen Kombinationen möglich sind, folgt, dass es insgesamt $2^{2^{10}}$ verschiedene Kombinationen von Blättern gibt.

Als nächstes überlegen wir uns, wie viele innere Knoten es gibt. Da wir die unterste Ebene des Baumes (also die Blätter) bereits abgearbeitet haben, ist die Tiefe des Baumes nun 9. Da es wie oben jeweils 2 verschiedene Entscheidungen pro Knoten gibt, erhält man $2^0 + 2^1 + ... + 2^9 = 2^{10} - 1$ innere Knoten. In jedem inneren Knoten können nun approximiert 10 verschiedene Attribute getestet werden (natürlich würden Lernalgorithmen ein bereits getestetes Attribut in dem gleichen Ast des Baumes nicht wieder verwenden). Da diese jeweils

in allen möglichen Kombinationen auftauchen können, erhält man also $10^{2^{10}-1}$ verschiedene Möglichkeiten für die inneren Knoten.

Insgesamt kann man also abschätzen, dass man $10^{2^{10}-1} \cdot 2^{2^{10}}$ Entscheidungsbäume absuchen müsste, da jeder mögliche nicht balancierte und nicht vollbesetzte (Teil-) Baum in den abgeschätzten Bäumen enthalten ist. Nimmt man zum Beispiel einen Baum der im linken Teilbaum ab Tiefe 3 den Klassenwert "+" ausgibt, so müsste man in dem korrespondierenden voll besetzten, balancierten Baum zwar noch bis Tiefe 10 hinunterwandern, um bei einem Blatt anzukommen, würde aber ebenfalls immer den Klassenwert "+" erhalten (da man den Baum gewählt hat, der in dem betreffenden Teilbaum immer "+" ausgibt).

b) Wie viele (partielle) Entscheidungsbäume müssen maximal beim Verfahren des TDIDT untersucht werden?

Lösung: Wie man an der obigen Aufgabe gesehen hat, ist es nicht möglich einfach auf allen Entscheidungbäumen eine vollständige Suche auszuführen. Daher sind Lernverfahren nötig, um einen guten Entscheidungsbaum zu erstellen.

Beim Verfahren des TDIDT werden in jedem Knoten die 10 möglichen Attribute getestet und dann das ausgewählt, dass den besten Heuristikwert erhält. Geht man davon aus, dass man in jeder Ebene des Baumes immer alle Attribute zulässt, resultieren damit $2^{10} - 1$ Knoten (da die Blätter nicht mehr durchsucht werden müssen), bei denen jeweils eben 10 Attribute durchsucht werden müssen. Man erhält also $(2^{10}-1) \cdot 10$ verschiedene Bäume, die untersucht werden müssen.

Geht man davon aus, dass in der ersten Ebene 10 Attribute, in der 2. nur noch 9, in der 3. noch 8 usw. untersucht werden müssen (z.B. TDIDT mit Heuristik Gain), so erhält man $2^0 \cdot 10 + 2^1 \cdot 9 + 2^2 \cdot 8 + ... + 2^{10} \cdot 0 = 2012$ verschiedene Bäume. Die zugehörige Formel lautet:

$\sum_{i=0}^{m} x^i \cdot (y - i)$ mit $m \equiv Tiefe$, $x \equiv Anzahl\ an\ Attributwerten$ und $y \equiv Anzahl\ an\ Attributen$

c) Angenommen die Datenmenge bestünde aus 1000 Beispielen. Wie oft würde jedes Beispiel bei der TDIDT im Worst-Case angefaßt?

Lösung: Da jedes Beispiel genau in einem Blatt im Entscheidungsbaum zu finden ist (es können auch mehrere Beispiele in einem Blatt sein), gibt es für jedes Beispiel genau einen Pfad durch den Baum. Im schlimmsten Fall ist das Beispiel in einem Blatt auf der tiefsten Ebene. Da TDIDT in jedem Knoten herausfindet, welches Attribut der beste Test ist und es 10 Attribute gibt, wird ein Beispiel pro Knoten maximal 10 mal angefaßt. Wenn es sich auf der tiefsten Ebene befindet, passiert das genau 10 mal. Daraus folgt, dass jedes Beispiel im Worst Case $10 \cdot 10 = 100$ mal angefaßt wird.

d) Was würde sich bei a) und b) ändern, wenn

- jedes Attribut nicht 2, sondern 10 Attributwerte hätte?

Lösung: Generell hat man nun nicht mehr 2 verschiedene Testausgänge, sondern 10. Daher ändert sich die Anzahl von Kombinationen der Blätter zu $2^{10^{10}}$, da man immer noch 2 Klassen vorhersagt. Die Anzahl von Kombinationen der inneren Knoten ist in diesem Fall $10^{10^{10}-1}$. Daraus folgt, dass man nun bei vollständiger Suche insgesamt $10^{10^{10}-1} \cdot 2^{10^{10}}$ verschiedene Bäume testen müsste.

– die Attribute nicht nominal, sondern numerisch wären?

Lösung: Bei numerischen Attributen muss man analog zum Regel-Lernen maximal so viele neue Tests einführen, wie es Attributwerte gibt. Stellt man sich pro Attribut einen Zahlenstrahl vor, so könnte man immer bei Klassenwechseln eine Unterteilung vornehmen. Hat beispielsweise ein Attribut von $0, ..., 10$ immer den Klassenwert "+" und von $11, ..., 20$ den Wert "-", so bräuchte man hier nur abzufragen, ob der Wert des Attributs < 11 oder $>= 11$ ist.

Aufgabe 2

Gegeben sei folgende Beispielmenge:

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No
D15	Sunny	Mild	Normal	Weak	No

a) Erzeugen Sie einen Entscheidungsbaum mittels des Verfahrens ID3 (TDIDT mit Maß Gain).

Lösung: Zur Verwendung von *Gain* (Folie 14) müssen wir die *Entropie* (Folie 15) einzelner Beispielmengen, die sich durch die möglichen Tests ergeben, berechnen.

Am Anfang betrachten wir alle Beispiele. Mögliche Tests sind *Outlook*, *Temperature*, *Humidity* und *Wind*. Die Entropie der gesamten Beispiele (Menge *S*, davon 9 positiv, 6 negativ) berechnet sich wie folgt:

$$Entropie(S) = -\frac{9}{9+6}\log_2\left(\frac{9}{9+6}\right) - \frac{6}{9+6}\log_2\left(\frac{6}{9+6}\right) = 0.971$$

Entsprechend bestimmen wir die Entropien der Beispielmengen, die durch Aufteilung gemäß des jeweiligen Tests entstehen.

Test	Werte	p	n	Entropie
Outlook	Overcast	4	0	0,000
	Rain	3	2	0,971
	Sunny	2	4	0,918
Temperature	Cool	3	1	0,811
	Hot	2	2	1,000
	Mild	4	3	0,985
Humidity	High	3	4	0,985
	Normal	6	2	0,811
Wind	Strong	3	3	1,000
	Weak	6	3	0,918

Nun können wir die jeweiligen Gains berechnen:

- $Gain(S, Outlook) = 0,971 - \frac{4}{15} \cdot 0 - \frac{5}{15} \cdot 0,971 - \frac{6}{15} \cdot 0,918 =$ **0,28**
- $Gain(S, Temperature) = 0,971 - \frac{4}{15} \cdot 0,811 - \frac{4}{15} \cdot 1 - \frac{7}{15} \cdot 0,985 = 0,028$
- $Gain(S, Humidity) = 0,971 - \frac{7}{15} \cdot 0,985 - \frac{8}{15} \cdot 0,811 = 0,078$
- $Gain(S, Wind) = 0,971 - \frac{6}{15} \cdot 0,1 - \frac{9}{15} \cdot 0,918 = 0,02$

Wir entscheiden uns für den Test mit maximalen Gain. Das wäre in diesem Fall der Test $Outlook$. Wir teilen demnach S folgendermaßen auf:

- $S_{Overcast} = \{D3, D7, D12, D13\}$
- $S_{Rain} = \{D4, D5, D6, D10, D14\}$
- $S_{Sunny} = \{D1, D2, D8, D9, D11, D15\}$

Für $S_{Overcast}$ wird kein weiterer Test benötigt, da diese Beispielmenge nur noch aus Beispielen einer Klasse besteht. Betrachten wir also S_{Rain}, für die nur noch die Tests *Temperature*, *Humidity* und *Wind* möglich sind. $Entropie(S_{Rain}) = 0,971$ ist uns bereits bekannt, wir benötigen also nur noch die Entropie der eben genannten Tests bezüglich S_{Rain}.

Test	Werte	p	n	Entropie
Temperature	Cool	1	1	1,000
	Hot	0	0	0,000
	Mild	2	1	0,918
Humidity	High	1	1	1,000
	Normal	2	1	0,918
Wind	Strong	0	2	0,000
	Weak	3	0	0,000

Nun können wir die jeweiligen Gains berechnen:

- $Gain(S_{Rain}, Temperature) = 0,971 - \frac{2}{5} \cdot 1 - \frac{3}{5} \cdot 0,918 = 0,020$
- $Gain(S_{Rain}, Humidity) = 0,971 - \frac{2}{5} \cdot 1 - \frac{3}{5} \cdot 0,918 = 0,020$

– $Gain(S_{Rain}, Wind) = 0,971 - \frac{2}{5} \cdot 0 - \frac{3}{5} \cdot 0 =$ **0,971**

Wir entscheiden uns für den Test $Wind$, der für eine Aufteilung in zwei Mengen, die nur aus positiven bzw. negativen Beispielen bestehen, sorgt. Es wird also kein weiterer Test benötigt.

Betrachten wir nun S_{Sunny} ($Entropie(S_{Sunny}) = 0,918$). Die Berechnung der möglichen Tests erfolgt analog zu S_{Rain}.

Test	Werte	p	n	Entropie
Temperature	Cool	1	0	0,000
	Hot	0	2	0,000
	Mild	1	2	0,918
Humidity	High	0	3	1,000
	Normal	2	1	0,918
Wind	Strong	1	1	1,000
	Weak	1	3	0,811

Damit erhalten wir die folgenden Gains: Nun können wir die jeweiligen Gains berechnen:

– $Gain(S_{Sunny}, Temperature) = 0,918 - \frac{1}{6} \cdot 0 - \frac{2}{6} \cdot 0 - \frac{3}{6} \cdot 0,918 =$ **0,459**
– $Gain(S_{Sunny}, Humidity) = 0,918 - \frac{3}{5} \cdot 1 - \frac{2}{5} \cdot 0,918 = 0,459$
– $Gain(S_{Sunny}, Wind) = 0,918 - \frac{2}{6} \cdot 1 - \frac{4}{6} \cdot 0,811 = 0,044$

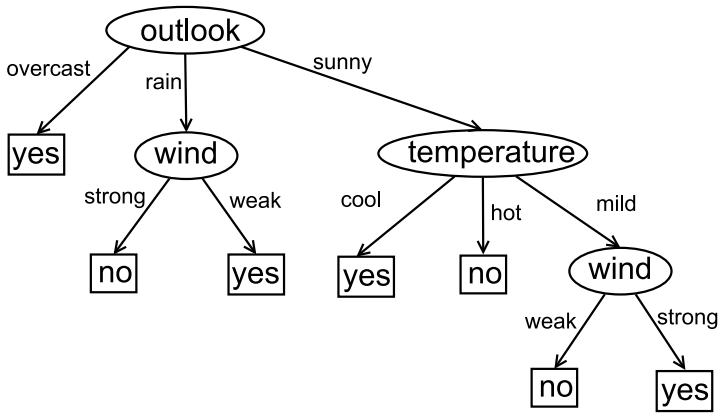
Wir entscheiden uns hier für den ersten Test mit maximalem Wert (zufälliges Wählen wäre auch möglich), d.h. für $Temperature$. Damit teilt sich S_{Sunny} wie folgt auf:

– $S_{Sunny,Cool} = \{D9\}$
– $S_{Sunny,Hot} = \{D1, D2\}$
– $S_{Sunny,Mild} = \{D8, D11, D15\}$

$S_{Sunny,Cool}$ und $S_{Sunny,Hot}$ bestehen nur aus positiven bzw. negativen Beispielen und müssen nicht weiter untersucht werden. Betrachten wir also als nächstes $S_{Sunny,Mild}$ ($Entropie(S_{Sunny,Mild}) = 0,918$). Mögliche Tests sind nur noch

Test	Werte	p	n	Entropie
Humidity	High	0	1	0,000
	Normal	1	1	1,000
Wind	Strong	1	0	0,000
	Weak	0	2	0,000

Wir entscheiden für den Test $Wind$, der die positiven von den negativen Beispielen trennt. Unser Entscheidungsbaum ist demnach fertig und sieht wie folgt aus:



b) Wiederholen Sie die Berechnungen für die Auswahl des Tests in der Wurzel mit den Maßen Information-Gain-Ratio und Gini-Index. Ändert sich etwas?

- GainRatio

  Lösung: GainRatio (Folie 20) berechnet sich aus *Gain* geteilt durch *SplitInformation* (Folie 20). Aus a) wissen wir bereits den Gain für die ersten Tests und müssen deshalb nur noch die *SplitInformation* berechnen. Im folgenden werden wir *SplitInformation* durch *SplitInfo* abkürzen

  - $SplitInfo(S, Outlook) = -\frac{4}{15}\log_2\frac{4}{15} - \frac{5}{15}\log_2\frac{5}{15} - \frac{5}{15}\log_2\frac{5}{15} = 1,566$
  - $SplitInfo(S, Temperature) = 2 \cdot \left(-\frac{4}{15}\log_2\frac{4}{15}\right) - \frac{7}{15}\log_2\frac{7}{15} = 1,53$
  - $SplitInfo(S, Humidity) = -\frac{7}{15}\log_2\frac{7}{15} - \frac{8}{15}\log_2\frac{8}{15} = 0,997$
  - $SplitInfo(S, Wind) = -\frac{6}{15}\log_2\frac{6}{15} - \frac{9}{15}\log_2\frac{9}{15} = 0,971$

  Damit erhalten wir die folgenden GainRatios:

  - $GainRatio(S, Outlook) = \frac{Gain(S,Outlook)}{SplitInfo(S,Outlook)} =$ **0,18**
  - $GainRatio(S, Temperature) = \frac{Gain(S,Temperature)}{SplitInfo(S,Temperature)} = 0,02$
  - $GainRatio(S, Humidity) = \frac{Gain(S,Humidity)}{SplitInfo(S,Humidity)} = 0,08$
  - $GainRatio(S, Wind) = \frac{Gain(S,Wind)}{SplitInfo(S,Wind)} = 0,02$

  Als erster Test wird *Outlook* ausgewählt. Die entstehenden Beispielmengen haben wir bereits in Aufgabe 1a) angegeben. $S_{Overcast}$ besteht nur aus positiven Beispielen und muß deshalb nicht weiter aufgeteilt werden. Betrachten wir nun als nächstes S_{Rain}. Die Gains haben wir bereits in a) berechnet, und die Berechnung der SplitInfo erfolgt analog zur obigen Berechnung. Wir erhalten also folgende GainRatios:

- $GainRatio(S_{Rain}, Temperature) = \frac{Gain(S_{Rain}, Temperature)}{SplitInfo(S_{Rain}, Temperature)}$
  $= \frac{0,02}{0,971} = 0,02$
- $GainRatio(S_{Rain}, Humidity) = \frac{Gain(S_{Rain}, Humidity)}{SplitInfo(S_{Rain}, Humidity)} = \frac{0,02}{0,971}$
  $= 0,02$
- $GainRatio(S_{Rain}, Wind) = \frac{Gain(S_{Rain}, Wind)}{SplitInfo(S_{Rain}, Wind)} = \frac{0,971}{0,971} = \mathbf{1}$

Der Test $Wind$ trennt die positiven von den negativen Beispielen. Wir müssen als keine weiteren Tests prüfen.

Schauen wir uns nun S_{Sunny}. Auch hier sind uns die Gains bereits bekannt, nur die SplitInfos müssen noch berechnet werden.

- $GainRatio(S_{Sunny}, Temperature) = \frac{Gain(S_{Sunny}, Temperature)}{SplitInfo(S_{Sunny}, Temperature)}$
  $= \frac{0,459}{1,459} = 0,31$
- $GainRatio(S_{Sunny}, Humidity) = \frac{Gain(S_{Sunny}, Humidity)}{SplitInfo(S_{Sunny}, Humidity)} = \frac{0,459}{1}$
  $= \mathbf{0,459}$
- $GainRatio(S_{Sunny}, Wind) = \frac{Gain(S_{Sunny}, Wind)}{SplitInfo(S_{Sunny}, Wind)} = \frac{0,044}{0,918} = 0,048$

Wir wählen also den Test $Humidity$ aus. Damit erhalten wir die folgenden Beispielmengen:

- $S_{Sunny,High} = \{D1, D2, D8\}$
- $S_{Sunny,Normal} = \{D9, D11, D15\}$

$S_{Sunny,High}$ müssen wir nicht betrachten, da sie nur aus negativen Beispielen besteht. Schauen wir uns als nächstes $S_{Sunny,Normal}$ an.

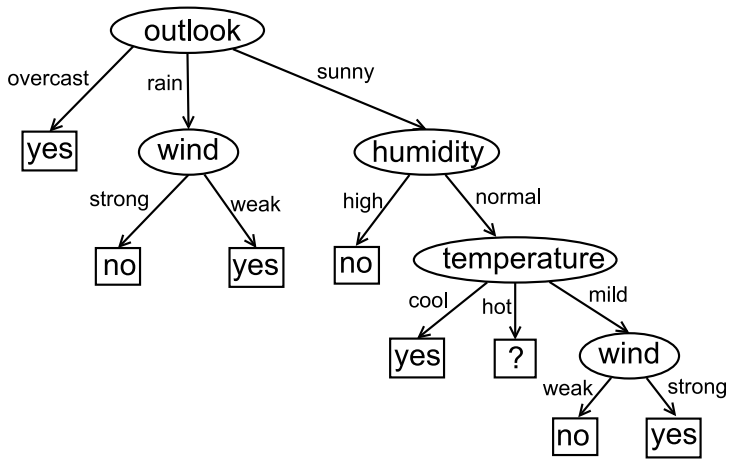
- $GainRatios(S_{Sunny,Normal}, Temperature)$
  $= \frac{Gain(S_{Sunny,Normal}, Temperature)}{SplitInfo(S_{Sunny,Normal}, Temperature)} = \frac{0,585}{0,959} = \mathbf{0,61}$
- $GainRatio(S_{Sunny,Normal}, Wind) = \frac{Gain(S_{Sunny,Normal}, Wind)}{SplitInfo(S_{Sunny,Normal}, Wind)}$
  $= \frac{0,459}{1} = 0,61$

Beide Tests haben das gleiche $GainRatio$, wir entscheiden uns wiederum für den ersten Test. Für diesen sind keine Werte für den Ausgang $Temperature = Hot$ bekannt, demnach müssen wir uns für eine Vorhersage in dem Blatt entscheiden. Entweder sagt man die in S am häufigsten vorkommende Klasse (positiv, da 9 positiv und 6 negativ) oder die in $S_{Sunny,Normal}$ am häufigsten vertretene Klasse (negativ, da 2 positiv und 4 negativ) voraus. Da man nicht weiß, welche dieser Lösungen besser funktioniert, werden wir in unserem Baum in dieses Blatt ein ? eintragen. Bei der Implementierung eines Entscheidungsbaumlerners würde man sich jedoch für eine der beiden Methoden entscheiden.

- $S_{Sunny,Cool,Normal} = \{D9\}$
- $S_{Sunny,Hot,Normal} = \emptyset$

– $S_{Sunny,Mild,Normal} = \{D11, D15\}$

Wir betrachten nur noch $S_{Sunny,Mild,Normal}$, da $S_{Sunny,Cool,Normal}$ nur aus positiven Beispielen besteht. Bei $S_{Sunny,Mild,Normal}$ können wir nur noch $Wind$ testen. Damit ist unser Baum fertig und sieht wie folgt aus:



- Gini-Index

  Lösung: Wir verwenden nun zur Auswahl des Test den *Gini-Index* (Folie 21). Hierfür berechnen wir zuerst $g_i(S_{Test})$ für jeden Tests $Outlook$, $Temperature$, $Humidity$ und $Wind$.

Test	Werte	p	n	$g_i(S_{Test})$
Outlook	Overcast	4	0	0,000
	Rain	3	2	0,480
	Sunny	2	4	0,444
Temperature	Cool	3	1	0,375
	Hot	2	2	0,500
	Mild	4	3	0,490
Humidity	High	3	4	0,490
	Normal	6	2	0,375
Wind	Strong	3	3	0,500
	Weak	6	3	0,444

  Damit erhalten wir die folgenden *Gini-Indizes*:

  – $gini(S, Outlook) = \frac{4}{15} \cdot 0 + \frac{5}{15} \cdot 0,48 + \frac{6}{15} \cdot 0,444 =$ **0,338**
  – $gini(S, Temperature) = \frac{4}{15} \cdot 0,375 + \frac{4}{15} \cdot 0,5 + \frac{7}{15} \cdot 0,49 = 0,462$
  – $gini(S, Humidity) = \frac{7}{15} \cdot 0,49 + \frac{8}{15} \cdot 0,375 = 0,429$

– $gini(S, Wind) = \frac{6}{15} \cdot 0,5 + \frac{9}{15} \cdot 0,444 = 0,467$

Im Gegensatz zu $Gain$ und $GainRatio$ wählen wir nun den Test mit dem geringsten $Gini - Index$ aus. Dies ist wiederum $Outlook$. $S_{Overcast}$ müssen wir wie bereits erwähnt nicht weiter betrachten. Schauen wir uns also zunächst S_{Rain} an und berechnen für alle möglichen Tests $g_i(S_{Test})$.

Test	Werte	p	n	$g_i(S_{Test})$
Temperature	Cool	1	1	0,500
	Hot	0	0	0,000
	Mild	2	1	0,444
Humidity	High	1	1	0,500
	Normal	2	1	0,444
Wind	Strong	0	2	0,000
	Weak	3	0	0,000

Damit erhalten wir die folgenden *Gini-Indizes*:

– $gini(S_{Rain}, Temperature) = 0,467$
– $gini(S_{Rain}, Humidity) = 0,467$
– $gini(S_{Rain}, Wind) = \mathbf{0}$

Wir wählen den Test $Wind$ aus, der die positiven von den negativen Beispielen trennt.

Schauen wir uns nun S_{Sunny} an. Da die Berechnungen analog zu den vorherigen erfolgen, werden wir sie ab jetzt nicht weitererläutern.

Test	Werte	p	n	$g_i(S_{Test})$
Temperature	Cool	1	0	0,000
	Hot	0	2	0,000
	Mild	1	2	0,444
Humidity	High	0	3	0,000
	Normal	2	1	0,444
Wind	Strong	1	1	0,500
	Weak	1	3	0,375

Damit erhalten wir die folgenden *Gini-Indizes*:

– $gini(S_{Sunny}, Temperature) = \mathbf{0,222}$
– $gini(S_{Sunny}, Humidity) = 0,222$
– $gini(S_{Sunny}, Wind) = 0,417$

Wir wählen also den Test $Temperature$ aus. Wir müssen nur die Menge $S_{Sunny,Mild}$ auf weitere Tests prüfen.

Test	Werte	p	n	$g_i(S_{Test})$
Humidity	High	0	1	0,000
	Normal	1	1	0,500
Wind	Strong	1	0	0,000
	Weak	0	2	0,000

Damit erhalten wir die folgenden *Gini-Indizes*:

- $gini(S_{Sunny,Mild}, Humidity) = 0,333$
- $gini(S_{Sunny,Mild}, Wind) = \mathbf{0}$

Als letzten Test wählen wir $Wind$ aus und erhalten den Baum aus a).

c) Ersetzen Sie das Beispiel D1 durch:

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	?	Hot	High	Weak	No

? steht hier für einen unbekannten/fehlenden Attributwert.

Überlegen Sie sich, wie man unbekannte/fehlende Attributwerte behandeln könnte.

Lösung: Unter anderem gibt es zwei Möglichkeit dieses Problem zu behandeln. Bei der ersten Möglichkeit ignorieren wir bei der Bewertung eines Tests alle Beispiele, deren Testausgang wir nicht kennen. Einerseits ist diese Vorgehensweise leicht zu implementieren, andererseits kann dies bei Datensätzen mit vielen fehlenden Werten zu kleinen Trainingsmengen führen, da viele Beispiel ignoriert werden.

Bei der zweiten Möglichkeit werden Beispiele, deren Attributwerte nicht vollständig sind, bei der Bewertung von den betroffenen Tests prozentual (im Verhältnis des Auftretens der einzelnen Attributwerte) mit einbezogen. Prozentuale Anteile dieser Beispiele werden dann immer wieder in Richtung der Blätter durchgereicht. Einerseits benötigt diese Problembehandlung eine komplexere Implementierung, andererseits gehen keine Daten “verloren”.