Textkategorisierung : Entscheidungsbäume

Kay-Michael Würzner Sven Brüssow

4. Juni 2003

Entscheidungsbäume

- Ein Entscheidungsbaum ist eine spezielle Datenstruktur um **Instanzen** zu klassifizieren. Innere Knoten repräsentieren **Attribute** einer **Klasse**, und Kanten repräsentieren Attributwerte.
- Eine Instanz wird klassifiziert indem, vom Wurzelknoten beginned, *top down* die Attributknoten entsprechend der möglichen Werte getestet werden und schließlich ein Terminalknoten erreicht wird.
- Ein Entscheidungsbaum repräsentiert allgemein eine **Disjunktion von Konjunktionen** von Beschränkungen auf die Attributwerte der Instanzen, mit anderen Worten jeder Pfad in einem Entscheidungsbaum vom Wurzelknoten zu einem Terminalknoten entspricht einer Konjunktion von Attributtests (*Humidity*,*Wind*) und der Baum selbst ist eine Disjunktion solcher Konjunktionen (1).

$$\begin{aligned} & (Outlook = Sunny \wedge Humidity = Normal) \\ \vee \quad & (Outlook = Overcast) \\ \vee \quad & (Outlook = Rain \wedge Wind = Weak) \end{aligned} \tag{1}$$

Entropie und Informationsgewinn

- Das Erlernen von Entscheidungsbäumen entspricht der *top down* Konstruktion des entsprechenden Baumes. Ein statistischer Test bestimmt dabei, wie gut ein Attribut allein die Trainingsdaten klassifiziert.
- Der aus der Informationstheorie bekannte Begriff der **Entropie** spielt dabei eine zentrale Rolle.

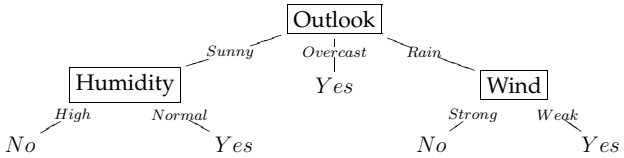

Abbildung 1: Ein Entscheidungsbaum für das Konzept *PlayTennis*

- Die Entropie ist ein gängiges Maß in der *Informationstheorie*.
- Die Entropie charakterisiert die (Un)Reinheit einer beliebigen Menge S von Beispielen.
- $p_\oplus$ ist der Anteil positiver Beispiele bestimmter Zielkonzepte in S.
- $p_\ominus$ ist der Anteil negativer Beispiele bestimmter Zielkonzepte in S.
- Die Entropie relativ zu eine boolschen Klassifizierung errechnet sich mit (2)
- Für nicht binäre sondern für c-wertige Klassifikationen errechnet sich die Entropie nach (3).
- Der Logarithmus in (3) ist weiterhin zur Basis 2, da die Entropie ein Maß für die erwartete Kodierungslänge gemessen in *bits* ist.

$$Entropy(S) = -p_\oplus \log_2 p_\oplus - p_\ominus \log_2 p_\ominus \tag{2}$$

$$Entropy(S) = \sum_{i=1}^{c} -p_i \log_2 p_i \tag{3}$$

- Die statistische Eigenschaft, die misst, wie gut ein Attribut die Trainingsdaten gemäß ihrer Zielklassifikation separiert und damit die Entropie reduziert nennt man Informationsgewinn (***information gain***).
- Der Informationsgewinn (*information gain*) gibt somit den Beitrag eines Attributs zur Entscheidungsfindung wieder.
- Der Informationsgewinn ist das Maß der Effektivität eines Attributs.
- Dieses Maß ist die Reduktion in der Entropie hervorgerufen durch die Partitionierung der Daten gemessen an dem entsprechenden Attribut.

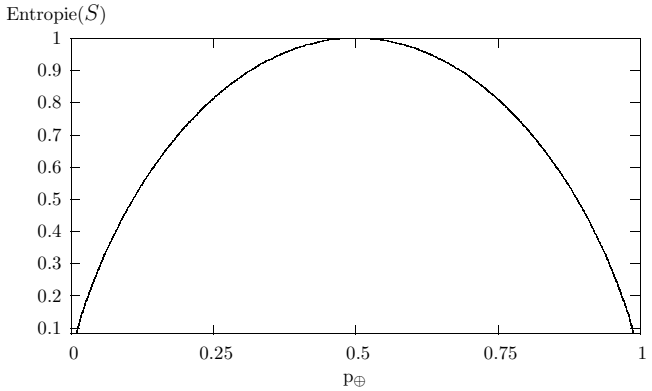

Abbildung 2: Die Entropiefunktion

$$Gain(S) \equiv Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v) \tag{4}$$

- Der erste Term in (4) ist die Entropie der gesamten Menge S.
- Der zweite Term in (4) ist der erwartete Wert der Entropie nachdem S durch das Attribut A partitioniert wurde.
- Der zweite Term ist somit die Summe der Entropien einer jeden Teilmenge S_v, die durch $\frac{|S_v|}{|S|}$ gewichtet werden.
- $Gain(S)$ ist die erwartete Reduktion in der Entropie hervorgerufen durch das Wissen um den Wert eines bestimmten Attributs.
- $Values(A)$ ist die Menge aller möglichen Werte für das Attribut A.
- S_v ist die Teilmenge von S für die das Attribut A den Wert v hat (5).

$$S_v = \{s \in S | A(s) = v\} \tag{5}$$

Bestimmung des besten Attributs und $ID3$

- Zur Bestimmung des Wurzelknoten-Attributs wird für jedes Attribut der *information gain* bestimmt.
- Das Attribut mit dem höchsten *information gain* wird der Wurzelknoten des Entscheidungsbaums.
- Das Wurzelknoten-Attribut ist am besten geignet, die Trainingsbeispiele in der folgenden Tabelle zu klassifizieren.

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

- Der Informationsgewinn, der die 14 Trainingsbeispiele in Abhängigkeit des Attributs $Wind$ sortiert, läßt sich folgendermaßen berechnen:

$$\begin{aligned}
Values(Wind) &= Weak, Strong \\
S &= [9+, 5-] \\
S_{Weak} &\leftarrow [6+, 2-] \\
S_{Strong} &\leftarrow [3+, 3-] \\
Gain(S, Wind) &= Entropy(s) - \sum_{v \in \{Weak, Strog\}} \frac{|S_v|}{|S|} Entropy(S_v) \\
&= Entropy(S) - (8/14) Entropy(S_{Weak}) - (6/14) Entropy(S_{Strong}) \\
&= 0.940 - (8/14) 0.811 - (6/14) 1.00 \\
&= 0.048
\end{aligned}$$

- Für die alle Attribute ergeben sich die folgenden Werte:

$$\begin{aligned} Gain(S, Outlook) &= 0.246 \\ Gain(S, Humidity) &= 0.151 \\ Gain(S, Wind) &= 0.048 \\ Gain(S, Temperature) &= 0.029 \end{aligned}$$

- Der $ID3$ **Algorithmus** benutzt Entropie und *information gain* um den Baum aufzubauen.
- Das Attribut $Outlook$ liefert den höchsten Informationsgewinn und somit die beste Vorhersage für das **Zielattribut** $PlayTennis$ und wird somit zum Wurzelknoten-Attribut (-Test), bezogen auf die Trainingsbeispiele.
- Für jeden seiner möglichen Werte ($Sunny$, $Overcast$, $Rain$) wird ein neuer Zweig eingefügt.
- Der Wert der Entropie für das Attribut $Outlook$ ist Null. Das bedeutet, dass der Test mit dem Attribut $Outlook$ und dessen Wert $Overcast$, in diesem Fall, zu einem Terminalknoten mit der **Klassifizierung** $PlayTennis = yes$ führt.
- An allen Attributknoten, an denen die Entropie nicht Null ist, muß der Entscheidungsbaum unterhalb dieser Knoten weiter untersucht werden.
- Der Prozeß des Auswählens von Attributen wird für jeden Folgeknoten wiederholt.
- Bereits in den Baum aufgenommene Attribute werden dabei nicht mehr berücksichtigt, da jedes Attribut nur einmal im Baum vorkommen darf.
- Der Prozeß endet für jeden Terminalknoten, wenn entweder
  - Jedes der Attribute entlang des Pfades plaziert wurde,

  oder
  - Die Trainingsbeispiele, die mit diesem Terminalknoten assoziiert sind, alle den selben Wert für das entsprechende Zielattribut haben, also die **Entropie Null** ist (s.o. bei $Outlook = Overcast$).

Eigenschaften des $ID3$ Algorithmus

- $ID3$ ist eine induktive Lernmethode bzw. induktiver Algorithmus.
- $ID3$ durchsucht einen Raum von Hypothesen (*space of hypothesises*) nach einer, die seinen Trainigsdaten am besten entspricht.
- Der Hypothesenraum, der von $ID3$ durchsucht wird, ist die Menge der möglichen Entscheidungsbäume.
- Die **Evaluierungsfunktion**, die $ID3$ leitet, ist das Maß des Informationsgewinns (*information gain*).
- Der Suchraum von $ID3$, also der Hypothesensuchraum aller Entscheidungsbäume, ist ein **kompletter Raum von endlichen diskret-wertigen (*discret-valued*)** Funktionen, relativ zu den vorhandenen Attributen.
- Da jede Funktion dieser Art als ein Entscheidungsbaum repräsentiert werden kann, vermeidet $ID3$ das Hauptisiko von Methoden, die unvollständige (*incomplete*) Hypothesen Suchräume durchsuchen: nämlich dass der Hypothesensuchraum die Zielfunktion nicht enthält.
- $ID3$ hält immer nur eine Hypothese aufrecht.
- Dadurch kann $ID3$ allerdings nicht bestimmen, wie viele alternative Entscheidungsbäume mit den vorhandenen Trainingsdaten konsistent sind.
- $ID3$ macht **kein *Backtracking***.
- Dadurch können **Entscheidungen lokal, aber nicht global optimal** sein.
- Der Vorteil durch das Heranziehen statistischer Eigenschaften wie *information gain* ist, dass die resultierende Suche weniger anfällig für Fehler in den Trainigsdaten ist.
- $ID3$ kann z.B. durch Modifizieren seiner Terminationsbedingungen erweitert werden, um Hypothesen zu akzeptieren, die ungewöhnlicher Weise in den Trainingsdaten vorkommen (*that imperfectly fit the training data*).

Inductive Bias

- *Inductive Bias* = Induktive Tendenz?
- *Inductive bias* ist die Menge der Annahmen, die, zusammen mit den Trainingsdaten, deduktiv die den folgenden Instanzen zugewiesenen Klassifizierungen rechtfertigen.

- Den *inductive bias* von $ID3$ zu beschreiben bedeutet, die Basis zu beschreiben, aufgrund der eine konsistente Hypothese einer andereren ebenfalls konsistenen Hypothese vorgezogen wird.
- $ID3$ entscheidet sich für den ersten akzeptablen Baum, auf den er in seiner Suche trifft, mit anderen Worten:
  - $ID3$ zieht kürzere Bäume den Längeren vor.
  - $ID3$ favorisiert die Bäume, die Attribute mit dem höchsten Informationsgewinn dem Wurzelknoten am nächsten plazieren.
- Durch die Anwendung einer ***greedy heuristic search*** werden die kürzesten Bäume gefunden, ohne den ganzen Raum *breadth first* absuchen zu müssen.
- $ID3$ findet nicht immer den kürzesten Baum, da er die Bäume favorisiert, die die Attribute mit dem höchsten Informationsgewinn wurzelknotennah plazieren (s.o).

Overfitting

- $ID3$ erweitert jeden Zweig des Baumes gerade tief genug, um die Trainingsdaten genau zu klassifizieren.
- Bei aus der Reihe fallenden Trainingsdaten oder einer zu geringen Anzahl von Trainingsbeispielen kann dies zu Schwierigkeiten bei der Bestimmung der Zielfunktion führen.
- **Definition: Gegeben ist ein Hypothesensuchraum H, eine Hypothese $h \in H$ *overfits* die Trainingsdaten wenn eine alternative Hypothese $h' \in H$ existiert, so dass h weniger Fehler als h' auf den Trainingsbeispielen hat, h' aber weniger Fehler als h auf dem gesamten Suchraum hat.**
- Die Genauigkeit des Entscheidungsbaums wächst für die Trainingsbeispiele monoton, während sie für Testbeispiele, die nicht in der Trainingsmenge enthalten sind, zunächst ansteigt, dann aber wieder abnimmt.
- **Beispiel: *Noisy data***
  - $\langle Outlook = Sunny, Temp = Hot, Hum = Normal, Wind = Strong \rangle$

    Anstatt das aus dem Rahmen fallende Beispiel zu ignorieren, produziert $ID3$ einen komplexeren Baum als nötig.
  - Der einfachere Baum zu h' paßt nicht mehr zu den Trainigsdaten.
  - Der komplexere Baum h allein gibt die Zielfunktion wieder.

- **Eine mögliche Lösung: *Post-Pruning***
  - *Post pruning* ist eine Form des *Backtracking*.
  - Neben der Trainingsmenge wird noch eine Validierungsmenge herangezogen.

Was ist Textkategorisierung?

- Problem:
  - große Sammlung von Texten S
  - gehört Text t zu einer Kategorie c ?
- Idee:
  - Korpus in dem die einzelnen Texte kategorisiert sind
  - Auswahl von Eigenschaften, die die Kategorie charakterisieren
  - bei Texten Wörter und Zeichen
  - es wird ein Klassifikationsmodell erstellt
  - getestet wird mit dem $Testkorpus$
- Anwendung:
  - unkategorisierte Texte könne nun leicht ihrer $wahrscheinlichsten$ Kategorie zugeordnet werden
  - Bsp. Newsticker: eingehende Nachrichten der Agenturen werden klassifiziert und den entsprechenden Redaktionen zugeordnet
- Schwierigkeiten:
  - Arbeit mit Wahrscheinlichkeiten
  - bei Kategorisierung anhand von Eigenschaften, Auswahl dieser Eigenschaften große Kunst
  - meist relativ problemspezifisch

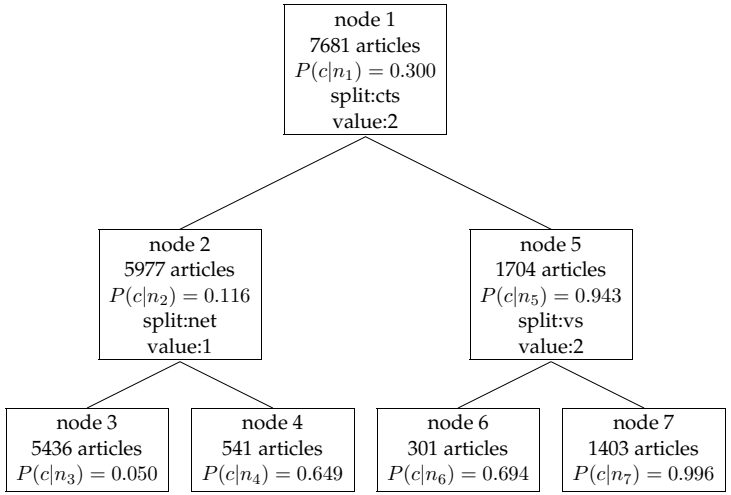

Abbildung 3: Entscheidungsbaum, der festlegt ob ein Dokument zur Kategorie "earnings" gehört.

Textkategorisierung mit Entscheidungsbäumen

Überblick

→ ein mögliches Klassifikationsmodell sind Entscheidungsbäume:

- dabei wird ein Entscheidungsbaum für die Kategorie, zu der zugeordnet werden soll, erstellt
- Voraussetzung bereits kategorisierter Korpus
  - dieser wird in Trainings- und Testkorpus geteilt
  - anhand der Daten des Trainingskorpus wird der Baum trainiert
  - mit dem Testkorpus wird dieser dann getestet
- beginnend beim Wurzelknoten werden die ausgewählten Eigenschaften abgefragt und die Texte somit eingeteilt
- an jedem Knoten wird angegeben:
  - Anzahl der Texte am Knoten
  - Wahrscheinlichkeit zur Kategorie zu gehören
  - Wort mit dem weiter geteilt wird
  - Gewicht des Teilungskriteriums

<REUTERS NEWID="11">
<DATE>26-FEB-1987 15:18:59.34</DATE>
<TOPICS><D>earn</D> </TOPICS>
<TEXT>
<TITLE>COBANCO INC <CBCO> YEAR NET </TITLE>
<DATELINE> SANTA CRUZ, Calif., Feb 26 - </DATELINE>
<BODY>Shr 34 cts vs 1.19 dlrs
Net 807,000 vs 2,858,000
Assets 510.2 mln vs 479.7 mln
Deposits 472.3 mln vs 440.3 mln
Loans 299.2 mln vs 327.2 mln
Note 4th qtr not available. Year includes 1985
extraordinary gain from tax carry forward of 132,000 dlrs,
or five cts per shr.
Reuter
</BODY></TEXT>
</REUTERS>

Abbildung 4: Beispiel aus dem $Reuters$ Korpus, Text der Kategorie $earnings$

- an den Blättern stehen keine Teilungskriterien, es soll ja nicht weiter geteilt werden

Auswahl der charakteristischen Wörter

- Entscheidung ob ein Text zu einer Kategorie gehört, benötigt Kriterien
- Auswahl von 20 Wörtern die für die Textsorte charakteristische sind
- basierend auf den Wörtern mit dem höchsten X^2 Wert für die Kategorie in der Trainingsmenge
  - der χ^2 Test findet Zusammenhänge zwischen Wörtern und Kategorien in Texten
  - dazu wird für jedes Wort eine Tabelle angelegt:

	1	2	3
$earnings$			
$\neg earnings$			

  - in die Zellen wird dann die Anzahl der Texte in denen das Wort $1\times$, $2\times, \ldots$ vorkommt
  - mit $X^2 = \sum_{i,j} \frac{(o_{ij}-E_{ij})^2}{E_{ij}}$ berechnet man den X^2 Wert

$Word\ w_j$	$Term\ weight\ s_{ij}$	$Classification$
vs		
mln		
cts		
;	$\vec{x} = \begin{pmatrix} 5 \\ 5 \\ 3 \\ 3 \\ 3 \\ 4 \\ 0 \end{pmatrix}$	$c = 1$
&		
000		
loss		

Tabelle 1: Ausschnitt aus dem Representationsmodell für Dokument 11(4)

- mit diesen 20 Wörtern kann jeder Text als Vektor von $K = 20$ ganzen Zahlen, $\vec{x_j} = (s_{1j}, \ldots, s_{Kj})$, dargestellt werden
  - $s_{ij} = round(10 * \frac{1+\log(tf_{ij})}{1+\log(l_j)})$
  - t_{ij} ... Anzahl der Vorkommen von Term i im Dokument j
  - l_j ... Länge des Dokuments j
- das ganze ist dann ein $Representations - Modell$

Training des Entscheidungsbaumes

- der Wurzelknoten enthält zunächst alle Texte des Trainingskorpus und die Wahrscheinlichkeit, dass ein Text zur Kategorie gehört
- nun benötigen wir ein Kriterium um die Texte aufzuteilen - $splitting\ criterion$
  - dazu wird der $information\ gain$ genutzt
  - also das Wort aus dem Vektor auswählen, das die Entropie am Knoten am meisten reduziert
  - für jeden Folgeknoten wird dies wiederholt
- wenn das $stopping\ criterion$ erreicht ist, ist der Vorgang abgeschlossen
  - als $stopping\ criterion$ können verschiedene Größen fungieren
  - es wäre z.B möglich aufzuhören wenn alle 20 Wörter benutzt wurden
  - oder die Entropie nicht mehr entscheidend reduziert werden kann
  - oder die Wahrscheinlichkeiten an den Blättern nahe 0 oder 1 gehen

Overfiting - Pruning

- normalerweise wird im ersten Ablauf des "Baum-Trainings" ein sehr großer Baum erzeugt → bestes Ergebnis für Trainigskorpus
- diese großen Bäume $overfit$en die Trainingsmenge
  - $Overfitting$ passiert wenn beim klassifizieren Entscheidungen auf Grund zufälliger Eigenschaften des Trainingskorpus getroffen werden
  - beispielsweise enthält nur ein Dokument die Wörter $dlrs$ und pct und dieses gehört zur $earnings$-Kategorie
  - nun kann es passieren, dass später alle Dokumente in denen die beiden Wörter vorkommen der Kategorie zugeordnet werden
- dieses wird vermieden indem der Baum nachträglich beschnitten wird - $pruning$ - um eine vernünftige, aussagekräftige Größe zu erhalten
  - dabei wird bei jedem $pruning$ der am wenigsten hilfreiche Knoten entfernt
  - das wird so lang wiederholt bis kein Knoten mehr übrig ist
  - die so entstandene Menge von Bäumen wird mit einem unabhängigen $validation\ set$ getestet, der kleinste Baum mit dem akkuratesten Ergebnis wird ausgewählt
- unschön ist dabei, dass wir ungefähr 20% vom Trainingskorpus als Überprüfungsmenge benutzen müssen und die Auswahl des Baumes dann im Prinzip auch nur von diesen 20% abhängt
- "schlauere" Methode: n-fold $cross - validation$
  - dabei wird der Trainingskorpus in n Teile geteilt
  - der Baum wird mit n-1 Teilen trainiert, der n. Teil ist das $validation\ set$ auf dessen Basis gepruned wird
  - diesen Vorgang wiederholt man n-1 mal, wobei jeder Teil einmal als $validation\ set$ dient
  - die Durchschnittsgröße der beschnittenen Bäume dient uns als optimale Größe für den Entscheidungsbaum
  - nun wird der Baum mit dem kompletten Trainingskorpus trainiert und auf diese Größe zurecht gestutzt

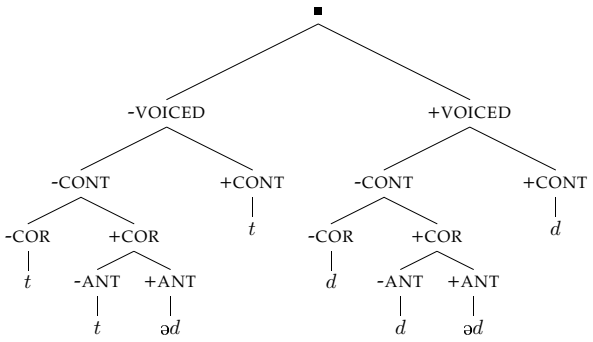

Abbildung 5: Ein Beispiel wie Entscheidungsbäume Daten phonologischer Regeln ineffizient benutzen.

NLP und Entscheidungsbäume

- für das Erfassen von natürlich sprachlichen Phänomen bzw. Generalisierungen bieten sich einfachere Methoden an
- Entscheidungsbäume teilen die Trainingsmenge in immer kleinere Untermengen auf
- dies macht korrekte Generalisierungen schwerer, weil die Datenmenge für verlässliche Aussagen zu klein ist
- und falsche Generalisierungen werden wahrscheinlicher, wegen zufälliger Regularitäten in kleineren Mengen
- Pruning löst dieses Problem bis zu einem gewissen Grad, andere Methoden können lingustische Regularitäten besser erfassen, indem sie alle Merkmale und Attribute gleichzeitig betrachten
- 5 zeigt ein Beispiel wie ein Entscheidungsbaum die Endungsregel im Englischen Past erfassen würde

Literatur

[Mit97] Tom Mitchel. *Machine Learning*. McGraw Hill, 1997.

[MS99] Christopher D. Manning and Hinrich Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, Massachusetts, 1999.