Einführung: Konzeptlernen

Caroline Sporleder

Computational Linguistics
Universität des Saarlandes

Sommersemester 2011

21.04.2011

Einführung

CL in den letzten 30 Jahren

Vor 1990/95:

- manuell erstellte Regeln, z.B. zum Auflösen von Pronomen
- Anwendung auf einen klar umrissenen, relativ kleinen Problembereich (z.B. Auflösen von Pronomen der 3. Person, bei denen der Antezedent innerhalb der 3 vorangehenden Sätze vorkommt)

Seit 1990/95:

- mehr und mehr Entwicklung von Systemen, die ling. Regelmäßigkeiten selber lernen können

CL in den letzten 30 Jahren

Vor 1990/95:

- manuell erstellte Regeln, z.B. zum Auflösen von Pronomen
- Anwendung auf einen klar umrissenen, relativ kleinen Problembereich (z.B. Auflösen von Pronomen der 3. Person, bei denen der Antezedent innerhalb der 3 vorangehenden Sätze vorkommt)

Seit 1990/95:

- mehr und mehr Entwicklung von Systemen, die ling. Regelmäßigkeiten selber lernen können

Warum?

ML vs. manuell erstellte Regeln

- manuelles Erstellen von Regeln ist zeitaufwendig
- Erstellung von Ressourcen für ML kann auch aufwendig sein, aber Ressourcen können z.T. wiederverwendet werden
- manuelle Regeln sind of sehr präzise, haben aber nur einen geringen Abdeckungsgrad

$\Rightarrow$ **ingesamt ist ML meist robuster** (größerer Abdeckungsgrad, Fallbackstrategien etc.)

Was ist ML?

Mitchell, S. 2

"A computer program is said to **learn** from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E."

Ein Beispiel (1)

Max und die Vorlesung

- manchmal kommt Max zur “Daten- und Algorithmen”-Vorlesung, manchmal nicht
- ihr vermutet dahinter Regelmäßigkeit und wollt herausfinden, was die Regel ist
- über die nächsten Wochen notiert ihr eure Beobachtungen

Ein Beispiel (1)

Max und die Vorlesung

- manchmal kommt Max zur "Daten- und Algorithmen"-Vorlesung, manchmal nicht
- ihr vermutet dahinter Regelmäßigkeit und wollt herausfinden, was die Regel ist
- über die nächsten Wochen notiert ihr eure Beobachtungen

Beobachtungen

- **Doz**ent: Prof. Adalbert **A**lbers, Prof. Beatrice **B**ertel
- **The**ma: **D**aten, **A**lgorithmen
- **Wo**chentag: **D**ienstag, **F**reitag
- **We**tter: **s**onnig, **r**egnerisch

Ein Beispiel (2)

Beobachtungen nach ein paar Wochen

Num	Doz	The	Wo	We	Max da?
1	B	A	D	S	N
2	A	D	F	R	N
3	B	D	D	R	J
4	A	A	F	S	N
5	B	A	D	R	J

Was ist die Regel?

Ein Beispiel (3)

Max und die Vorlesung

- **Experience:**
- **Task:**
- **Performance Measure:**

Ein Beispiel (3)

Max und die Vorlesung

- **Experience:** die Beobachtungen
- **Task:**
- **Performance Measure:**

Ein Beispiel (3)

Max und die Vorlesung

- **Experience:** die Beobachtungen
- **Task:** Voraussage, ob Max da ist
- **Performance Measure:**

Ein Beispiel (3)

Max und die Vorlesung

- **Experience:** die Beobachtungen
- **Task:** Voraussage, ob Max da ist
- **Performance Measure:** Voraussagekorrektheit, z.B. **accuracy**

Konzeptlernen (1)

Terminologie

- die Gesamtheit der Beobachtungen sind die **Trainingsdaten** (die Menge X)
- jede Beobachtung ist eine **Instanz**
- eine Beobachtung besteht aus einer Menge von **Attribut-Wert Paaren** (AWP), z.B. Attribut=Dozent Wert=Adalbert Albers (Doz=A)
- zu jeder Instanz gehört auch ein **Ausgabewert**, ebenfalls in Form eines AWP, z.B. Max da?=J
- Instanzen, mit einem positiven Ausgabewert (=J) werden **positive Trainingsbeispiele** genannt, die anderen **negative Beispiele**
- eine Instanz kann als **Vektor** von Attributwerten dagestellt werden, z.B. <B,A,D,S> (auch **feature vector**)

Konzeptlernen (2)

Ziel

Finde eine Targetfunktion (c), die Instanzen korrekt auf (binäre) Ausgabewerte abbildet (=das Targetkonzept).

Formal

$c : X \rightarrow \{0, 1\}$

Konzeptlernen (3)

Hypothesenraum

Um die Targetfunktion zu finden, muss der Lernalgorithmus aus einer Menge von **Hypothesen**, H, die richtige, h, auswählen, d.h. $\forall x \in X h(x) = c(x)$
H wird auch **Hypothesenraum** genannt.

Repräsentation von Hypothesen

Eine Hypothese kann als Vektor dargestellt werden, der die möglichen Werte der Instanzattribute wie folgt eingrenzt:

- **?** der Wert des Attributs ist egal
- der Wert des Attributes ist festgelegt, z.B. **Adalbert Albers**
- $\emptyset$ das Attribut darf keinen Wert haben (ist aus formalen Gründen sinnvoll)

Beispiel

<A,?,?,S>

Hypothesen: Beispiel

Welche Hypothese(n) passen auf die positiven Instanzen?

Num	Doz	The	Wo	We	Max da?
1	B	A	D	S	N
2	A	D	F	R	N
3	B	D	D	R	J
4	A	A	F	S	N
5	B	A	D	R	J

Hypothesen: Beispiel

Welche Hypothese(n) passen auf die positiven Instanzen?

Num	Doz	The	Wo	We	Max da?
1	B	A	D	S	N
2	A	D	F	R	N
3	B	D	D	R	J
4	A	A	F	S	N
5	B	A	D	R	J

Lösung:

- <B,?,D,R>
- <B,?,?,R>

Konzeptlernen als Suche

Für unser einfaches Beispiel ist der Hypothesenraum endlich.

Naive Suchstrategie

1. Wähle eine zufällige Hypothese, h, aus.
2. Vergleiche h mit allen Trainingsinstanzen X,
   wenn h für alle $x \in X$ den korrekten Ausgabewert gibt, dann stop.
   sonst lösche h und gehe zurück zu 1

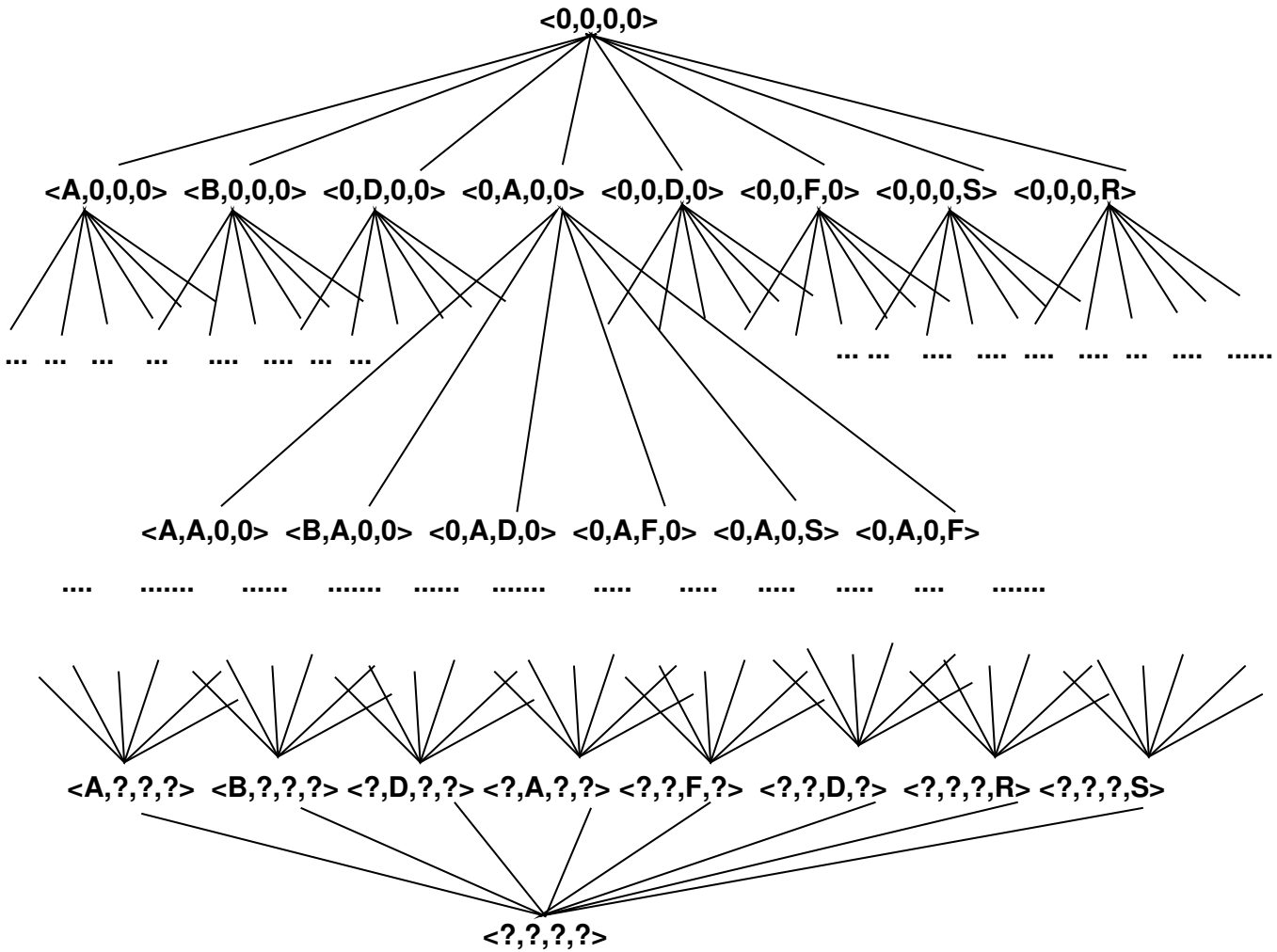
Problem

Hypothesenraum kann sehr schnell sehr groß werden
(hier 4*4*4*4=256).

Fällt euch eine bessere Suchstrategie ein?

Gerichtete Suche

Hypothesen können partiell geordnet werden



⇒ **erlaubt eine gerichtete Suche!**

Einfache Lernalgorithmen

Find-S

Idee:

finde die spezifischste Hypothese

1. beginne mit der spezifischsten Hypothese, h, $(< \emptyset, \emptyset, \emptyset, \emptyset >)$
2. generalisiere h für jedes positive Trainingsbeispiel, dass nicht abgedeckt wird

Find-S: Beispiel

Trainingsbeispiel	Hypothese h
	$< \emptyset, \emptyset, \emptyset, \emptyset >$

Find-S: Beispiel

Trainingsbeispiel	Hypothese h
	$< \emptyset, \emptyset, \emptyset, \emptyset >$
B D D R | J |	<B,D,D,R>

Find-S: Beispiel

Trainingsbeispiel					Hypothese h
					$< \emptyset, \emptyset, \emptyset, \emptyset >$
B	D	D	R	J	<B,D,D,R>
B	A	D	R	J	

Find-S: Beispiel

Trainingsbeispiel					Hypothese h
					$< \emptyset, \emptyset, \emptyset, \emptyset >$
B	D	D	R	J	<B,D,D,R>
B	A	D	R	J	<B,?,D,R>

Find-S: Beispiel

Trainingsbeispiel					Hypothese h
					$< \emptyset, \emptyset, \emptyset, \emptyset >$
B	D	D	R	J	<B,D,D,R>
B	A	D	R	J	<B,?,D,R>
A	D	F	R	N	

Find-S: Beispiel

Trainingsbeispiel					Hypothese h
					$< \emptyset, \emptyset, \emptyset, \emptyset >$
B	D	D	R	J	<B,D,D,R>
B	A	D	R	J	<B,?,D,R>
A	D	F	R	N	<B,?,D,R>

Find-S: Beispiel

Trainingsbeispiel					Hypothese h
					$< \emptyset, \emptyset, \emptyset, \emptyset >$
B	D	D	R	J	<B,D,D,R>
B	A	D	R	J	<B,?,D,R>
A	D	F	R	N	<B,?,D,R>
B	D	D	R	J	

Find-S: Beispiel

Trainingsbeispiel					Hypothese h
					$< \emptyset, \emptyset, \emptyset, \emptyset >$
B	D	D	R	J	<B,D,D,R>
B	A	D	R	J	<B,?,D,R>
A	D	F	R	N	<B,?,D,R>
B	D	F	R	J	<B,?,?,R>

Find-S und negative Beispiele

Solange die Trainingsdaten keine Fehler enthalten und c in H enthalten ist, haben nur positive Trainingsdaten einen Einfluss auf h (d.h., h wird durch negative Daten nie revidiert).

Nachteile von Find-S

Nachteile von Find-S

- Warum die spezifischste Hypothese?
- keine Garantie, dass Find-S die einzige Hypothese findet, die mit den Trainingsdaten konsistent ist.
- Was passiert, wenn es mehrere Hypothesen gibt, die gleich spezifisch und alle konsistent mit den Trainingsdaten sind?

- Inkonsistenzen in den Trainingsdaten werden ignoriert (da nur positive Beispiele einen Effekt haben) und können zu Fehlern führen

Nachteile von Find-S

- Warum die spezifischste Hypothese?
- keine Garantie, dass Find-S die einzige Hypothese findet, die mit den Trainingsdaten konsistent ist.
- Was passiert, wenn es mehrere Hypothesen gibt, die gleich spezifisch und alle konsistent mit den Trainingsdaten sind?
  $\Rightarrow$ Find-S ist abhängig von der Reihenfolge der Trainingsinstanzen
- Inkonsistenzen in den Trainingsdaten werden ignoriert (da nur positive Beispiele einen Effekt haben) und können zu Fehlern führen

Exkurs: Inkonsistente Trainingsdaten (1)

Num	Doz	The	Wo	We	Max da?
1	B	A	D	S	N
2	A	D	F	R	N
3	B	D	D	R	J
4	A	A	F	S	N
5	B	A	D	R	J

Exkurs: Inkonsistente Trainingsdaten (1)

Num	Doz	The	Wo	We	Max da?
1	B	A	D	S	N
2	A	D	F	R	N
3	B	D	D	R	J
4	A	A	F	S	N
5	B	A	D	R	J
6	**B**	**A**	**D**	**R**	**N**

Exkurs: Inkonsistente Trainingsdaten

Wodurch können Inkonsistenzen entstehen?

- es gibt einen weiteren Faktor, der einen Einfluss auf den Ausgabewert hat, aber der nicht durch die Attribute abgedeckt ist
  (z.B. ob Max am Abend vor der Vorlesung auf einer Party war)
  $\Rightarrow$ **die korrekte Hypothese ist nicht in H und kann nicht (vollständig) gelernt werden**
- es kann sein, dass der Wert eines Attributs bei einer Instanz falsch ist
  (vielleicht sagt Max nicht die Wahrheit, wenn man ihn fragt, ob er auf einer Party war)
  $\Rightarrow$ **die korrekte Hypothese kann trotzdem in H sein, aber möglicherweise nicht als solche erkannt werden**

$\Rightarrow$ **beide Faktoren kommen in der Praxis sehr häufig vor**

Version Spaces und Candidate-Elimination

Besser als Find-S:

Der Candidate-Elimination Algorithmus gibt **alle** Hypothesen zurück, die mit den Trainingsdaten konsistent sind.

Die Repräsentation aller dieser Hypothesen nennt sich **Version Space** (VS).

Der Version Space ist durch zwei Grenzmengen (boundary sets) repräsentiert:

- **S:** die Menge mit den spezifischsten Hypothesen im aktuellen VS
- **G:** die Mengen mit den unspezifischsten Hypothesen im aktuellen VS.

Version Spaces: Beispiel

Welche Hypothesen sind mit den folgenden Daten konsistent?
Welche Hypothesen sind in den boundary sets S und G enthalten?

Num	Doz	The	Wo	We	Max da?
1	A	D	F	R	N
2	B	D	D	R	J
3	B	A	D	R	J

Version Spaces: Beispiel

Welche Hypothesen sind mit den folgenden Daten konsistent?
Welche Hypothesen sind in den boundary sets S und G enthalten?

Num	Doz	The	Wo	We	Max da?
1	A	D	F	R	N
2	B	D	D	R	J
3	B	A	D	R	J

Boundary sets:

S: $\{< B, ?, D, R >\}$

G: $\{< B, ?, ?, ? >, <?, ?, D, ? >\}$

Der Candidate-Elimination Algorithmus (1)

For each training example d, do
    if d is a **positive example**
        remove from G any hypothesis inconsistent with d
        for each hypothesis s in S that is inconsistent with d
            remove s from S
            Add to S all minimal generalisations h of s such that
                1. h is consistent with d, and
                2. some member of G is more general than h
            remove from S any hypothesis that is more general than another hypothesis S
        end for
...

Der Candidate-Elimination Algorithmus (2)

...

else if d is a **negative** example

remove from S any hypothesis inconsistent with d

for each hypothesis g in G that is inconsistent with d

remove g from G

add the G all minimal specialisations h of g such that

1. h is consistent with d, and
2. some member of S is more specific than h

remove from G any hypothesis that is less general than another hypothesis in G

end for

end do

Candidate-Elimination: Beispiel

Num	Doz	The	Wo	We	Max da?

Boundary sets:

S: $\{< \emptyset, \emptyset, \emptyset, \emptyset >\}$

G: $\{<?, ?, ?, ? >\}$

Candidate-Elimination: Beispiel

Num	Doz	The	Wo	We	Max da?
1	B	D	D	R	J

Boundary sets:

Candidate-Elimination: Beispiel

Num	Doz	The	Wo	We	Max da?
1	B	D	D	R	J

Boundary sets:

S: $\{< B, D, D, R >\}$

G: $\{<?, ?, ?, ? >\}$

Candidate-Elimination: Beispiel

Num	Doz	The	Wo	We	Max da?
1	B	D	D	R	J
2	A	D	F	R	N

Boundary sets:

Candidate-Elimination: Beispiel

Num	Doz	The	Wo	We	Max da?
1	B	D	D	R	J
2	A	D	F	R	N

Boundary sets:

S: $\{< B, D, D, R >\}$

G: $\{< B, ?, ?, ? >, <?, ?, D, ? >\}$

Candidate-Elimination: Beispiel

Num	Doz	The	Wo	We	Max da?
1	B	D	D	R	J
2	A	D	F	R	N
3	B	A	D	R	J

Boundary sets:

Candidate-Elimination: Beispiel

Num	Doz	The	Wo	We	Max da?
1	B	D	D	R	J
2	A	D	F	R	N
3	B	A	D	R	J

Boundary sets:

S: $\{< B, ?, D, R >\}$

G: $\{< B, ?, ?, ? >, <?, ?, D, ? >\}$

Der Candidate-Elimination Algorithmus

Der gelernte Version Space wird zur korrekten Targetfunktion konvergieren, sofern

- es keine Fehler in den Trainingsdaten gibt
- die Targethypothese im Hypothesenraum vorhanden ist

Induktion vs. Deduktion

Find-S und Candidate-Elimination sind **induktive Lernverfahren**: aus konkreten Beobachtungen wird eine allgemeine Regel abgeleitet.
(Deduktion schließt aus einer allgemeinen Regel auf einen konkreten Fall)

Grundannahme des induktiven Lernens

Jede Hypothese, die aus einem genügend großen Trainingsset gelernt wurde und auf diesem Trainingsset die Targetfunktion approximiert, approximiert die Targetfunktion auch auf anderen, ungesehenen Instanzen.

Induktiver Bias (1)

Ein anderes Beispiel:

Unter welchen Umständen geht Max schwimmen?

Num	Sky	AirTemp	Humidity	Wind	Water	Forecast	schwimmt?
1	sunny	warm	normal	strong	cool	change	yes
2	cloudy	warm	normal	strong	cool	change	yes
3	rainy	warm	normal	strong	cool	change	no

Problem

Zielhypothese nicht im Hypothesenraum vorhanden.

$<?, \textit{warm}, \textit{normal}, \textit{strong}, \textit{cool}, \textit{change}>$ ist zu allgemein.

Wir bräuchten eine Disjunktion über *sunny* und *cloudy*.

Induktiver Bias (2)

Mögliche Lösung

Erweiterung des Hypothesenraums, so dass jede mögliche Targetfunktion abgedeckt ist.

$\Rightarrow$ der Hypothesenraum H muss jede mögliche Teilmenge der Menge der möglichen Instanzen enthalten (d.h. H ist die Potenzemenge von X).

Zum Beispiel

$\{< \mathit{sunny}, ?, ?, ?, ?, ? > \vee < \mathit{cloudy}, ?, ?, ?, ?, ? >\}$

Induktiver Bias (3)

Problem

In dem neuem Hypothesenraum ist es nicht möglich, über die gesehenen Beispiele hinaus zu abstrahieren!

Die S-Grenze ist jeweils die Disjunktion über die bisher gesehenen positiven Beispiele.

Die G-Grenze ist die Negation der Disjunktion der gesehenen negativen Beispiele.

Induktiver Bias (4)

Jedes induktive Lernverfahren hat einen **induktiven Bias**, der den Lerner zwingt, über die Trainingsdaten zu abstrahieren.
Der induktive Bias der gesehenen Konzeptlernverfahren besteht in der Repräsentation des Hypothesenraums (d.h. keine Disjunktionen über Attribut-Werte).

Ein Lernverfahren, dass sich lediglich die Trainingsinstanzen merkt und eine neue Instanz nur klassifizieren kann, wenn sie mit einer Trainingsinstanz identisch ist, nennt sich **rote learning** (kein echtes Lernen).

Zusammenfassung und Ausblick

Ihr solltet etwas gelernt haben über

Grundzüge maschinellen Lernens

- Instanzen
- Trainingsdaten
- Attribute und Werte
- Targetfunktionen

Einfache induktive Lernverfahren

- Repräsentation des Hypothesenraums
- Suche im Hypothesenraum (Find-S, Candidate-Elimination)
- die Notwendigkeit des induktiven Bias

Arten von Lernverfahren

Je nach Größe des Trainingssets kann man unterscheiden zwischen:

- **überwachtes Lernen** (supervised learning): großes Trainingsset
- **unüberwachtes Lernen** (unsupervised learning): kein Trainingsset (bzw. die 'Trainingsdaten' enthalten keine Ausgabewerte; unannotiertes Trainingsset)
- **teil-überwachtes Lernen** (semi-supervised learning): kleines Trainingsset plus große Menge unnannotierter Daten

In diesem Kurs...

...hauptsächlich überwachte Verfahren

- Decision Tree Learning: Lernen komplexer Regeln; informationstheoretische Grundlage
- Naive Bayes: statistisches Verfahren
- Memory Based Learning: Hypothesenbildung erst in der Testphase
- Genetische Algorithmen: Modellierung evolutionärer Strategien

teil-überwachte Verfahren

- Self-learning: ein Lerner, der vom eigenen Output lernt
- Co-Training: Zwei Lerner, die sich gegenseitig helfen

Meta-Lernen

- Ensemble Methods: Kombination verschiedener Verfahren