Moderne Methoden der KI: Maschinelles Lernen

Prof. Dr.Hans-Dieter Burkhard
Vorlesung Sommer-Semester 2007

Konzept-Lernen

Konzept-Lernen
Lernen als Suche
Inductive Bias

Konzept-Lernen: Problemstellung

Ausgangspunkt:
Konzept beschreibt Menge von Objekten (mit bestimmten Eigenschaften) aus einer umfassenderen Menge

X : Menge von Objekten x
Merkmalen $M_1,\ldots,M_m$ aus (endlichen) Wertebereichen $W_1,\ldots,W_m$
Objekte als Merkmalsvektoren $x = [x_1,\ldots,x_m]$

C: Konzept definiert durch Umfang, d.h. als Teilmenge von X: $C \subseteq X$
bzw. durch charakteristische Funktion $c_C: X \to \{0,1\}$

$$c_C(x) = \begin{cases} 1, & \text{falls } x \in C \\ 0, & \text{sonst} \end{cases}$$

Konzept-Lernen: Problemstellung

Konzept-Lernen:
Allgemeine Definition eines Konzeptes C anhand einiger positiver Beispiele ($x \in C$) und negativer Beispiele ($x \notin C$) erkennen

äquivalent:

Boolesche Funktion $c_C : X \to \{0,1\}$
anhand von Eingabe-/Ausgabepaaren lernen

Konzept-Lernen: Problemstellung

Lernen als Suche
Suchraum (Zustände/Operatoren):

Menge von Hypothesen
(hier für Konzepte $C \subseteq X$ bzw. Boolesche Fkten. $X \to \{0,1\}$)
evtl. nur eingeschränkte Hypothesen-Menge

Übergänge zwischen Hypothesen
(hier z.B. Halbordnung bzgl. Inklusion)

Suchstrategie: Anordnung ausnutzen
(hier z.B. umfassendere Hypothese suchen)

Konzept-Lernen: Notationen

Gesamtmenge: $X = W_1 x \ldots x W_m$,
Objekte als Merkmalsvektoren $x = [x_1,\ldots, x_m]$
Konzept: $C \subseteq X$ bzw. Charakt. Fkt. c_C : $c_C(x) = 1 \leftrightarrow x \in C$

korrekte Trainingsmenge D für Konzept C : Menge von Beispielen

$D = \{ [x, c_C(x)] \mid x \in X_D \}$ für eine Menge $X_D \subseteq X$
mit positiven Beispiele $[x, 1]$
und negativen Beispiele $[x, 0]$

Trainingsmenge D kann *fehlerbehaftet* ("verrauscht") bzgl. C sein:
fehlerhaftes Beispiel: $[x, c(x)]$ mit $c(x) \neq c_C(x)$

allgemein: $D = \{ [x, c(x)] \mid x \in X_D \}$
mit $c: X \to \{0,1\}$ (ohne Korrektheitsforderung)

Konzept-Lernen: Notationen

Hypothese h als Boolesche Funktion $h: X \to \{0,1\}$
beschreibt Konzept $C_h = \{ x \mid h(x) = 1 \}$

Lernziel bei Konzept C:

Hypothese h_C finden mit $\forall x \in X$: $h(x) = 1 \leftrightarrow x \in C$

Hypothese h ist konsistent mit Trainingsmenge D:

$\forall [x, c(x)] \in D : h(x) = c(x)$

i.a. gibt es viele mit D konsistente Hypothesen
(Problem der korrekten Verallgemeinerung)

Konzept-Lernen: Beispiel

Die **Menge** H der zugelassenen Hypothesen ist Entwurfsentscheidung.

Beispiel: Hypothesenmenge H_1

Hypothesen aus H_1 beschrieben durch verallgemeinerte Merkmalsvektoren $h = [h_1,\dots, h_m]$, mit $h_i \in \{ ? , x_i, \varnothing \}$
wobei der Eintrag h_i für das i-te Merkmal bedeutet:

$h_i = ?$: beliebige Ausprägung des i-ten Merkmals zugelassen
$h_i = x_i$: i-tes Merkmal muss Wert x_i haben
$h_i = \varnothing$: kein Wert zugelassen

$h(x) = h([x_1,\dots, x_m])= 1$ gdw. $\forall$ $i=1,\dots,m$: $h_i \in \{ ?, x_i \}$

Nicht alle Konzepte $C \subseteq X$ erfasst : $H_1 \not\subset 2^X$

General-to-Specific-Anordnung

Lernen als Suche basiert auf Anordnung des Suchraumes

"General-to-Specific"-Anordnung des Hypothesenraumes H durch Halbordnungsrelation für Boolesche Funktionen $h: X \to \{ 0,1\}$

h_1 allgemeiner oder gleich h_2 ($h_1 \supseteq h_2$)
gdw. $\forall x \in X : h_2(x) = 1 \to h_1(x)=1$

h_1 allgemeiner als h_2 ($h_1 \supset h_2$) gdw. $h_1 \supseteq h_2$ und $h_1 \neq h_2$

Im Beispiel H_1 : minimale Hypothese: $h = [\varnothing,\dots,\varnothing]$ ($C = \varnothing$)
maximale Hypothese: $h = [?,\dots, ?]$ ($C = X$)

Algorithmus *Find-S*

Idee: Mit speziellster Hypothese h beginnen.
Verallgemeinern, wenn ein positives Beispiel nicht überdeckt wird
(wenn $h(x) = 0$ für Beispiel $[x,1] \in D$)

Ausgangspunkt: Ein Hypothesenraum H
Beschränkungen der Merkmalswerte durch die Hypothesen
Bezeichnung: b_i = Beschränkung des i-ten Merkmals durch h

(z.B. H_1 mit $b_i \in \{ ?, x_i , \varnothing \}$)

Algorithmus *Find-S*

0. Initialisierung: h:= speziellste (minimale) Hypothese aus H

1. Für jedes positive Trainingsbeispiel $[[x_1,\dots, x_m] ,1] \in D$:
   Für jedes Merkmal $i=1,\dots,x$:
   - Falls x_i die Beschränkung b_i erfüllt: Nichts verändern.
   - Sonst: h minimal verallgemeinern durch
     Verallgemeinerung von b_i zu b'_i, so dass b'_i durch x erfüllt ist.

2. Hypothese h ausgeben.

Algorithmus *Find-S* (Fortsetzung)

FIND-S ignoriert negative Beispiele.

Wenn Hypothesen $h \in H$ jeweils Konjunktionen von Merkmalsbeschränkungen (b_i) sind
und Trainingsmenge korrekt ist, so gilt:

Find-S findet die spezifischste Hypothese h_{D+} aus H,
die mit den positiven Beispiele aus der Trainingsmenge D konsistent ist.
Falls Zielkonzept C in H enthalten ist, so ist die Hypothese h_{D+}
auch konsistent mit den negativen Beispielen aus D .

Einige Probleme für Lernsysteme

Problem-Diskussion (am Beispiel *Find-S*):

- Ist die gefundene Hypothese h **das** korrekte Zielkonzept C ? Im allg. mehrere mit Trainingsmenge D konsistente Hypothesen in H.
  - Algorithmus sollte Zuverlässigkeit von h angeben.
- Grund für Auswahl bei mehreren konsistenten Möglichkeiten?
  Hier: speziellste Hypothese,
  wegen Anordnung/Algorithmus (Design-Problem).
- Umgang mit fehlerhaften Beispielen:
  - Algorithmus sollte Inkonsistenzen in D erkennen und behandeln.
    (wird bei Find-S ignoriert: negative Beispiele ohne Einfluss)
- Umgang mit evtl. unterschiedlichen „besten" Elementen in H:
  - Backtracking vorsehen.

List-Then-Eliminate-Algorithmus

Version Space bzgl. H und D:

$VS_{H,D} := \{ h \in H \mid h \text{ konsistent mit } D \}$

= **alle** potentiellen Ergebnisse

List-Then-Eliminate-Algorithmus:

0. Initialisierung: $V := H$
1. Für jedes Trainingsbeispiel $[x, c(x)] \in D$:
   $V := V - \{ h \mid h(x) \neq c(x) \}$
2. Ergebnis: $VS_{H,D} := H$

zu komplex (Auflistung aller Hypothesen)

Candidate-Elimination-Algorithmus

Kompaktere Repräsentation für Version Space mittels Begrenzungsmengen:

$G := Max(VS_{H,D}) = \{ h \in VS_{H,D} \mid \neg \exists g \in VS_{H,D} : g \supset h \}$
$S := Min(VS_{H,D}) = \{ h \in VS_{H,D} \mid \neg \exists g \in VS_{H,D} : h \supset g \}$

G: allgemeinste Hypothesen aus $VS_{H,D}$
S: speziellste Hypothesen aus $VS_{H,D}$

Satz: $VS_{H,D} = \{ h \mid \exists g \in G \; \exists s \in S : g \supseteq h \supseteq s \}$
(Voraussetzung: G , S existieren)

Candidate-Elimination-Alg. arbeitet mit Begrenzungsmengen

Candidate-Elimination-Algorithmus

0. Initialisierung: $G := Max\ H$, $S := Min\ H$
1. Für jedes Trainingsbeispiel $d \in D$:
   Falls d positiv ($d = [x, 1]$):
   - $G := G - \{ g \mid g(x) \neq 1 \}$
   - Für alle $s \in S$ mit $s(x) \neq 1$:
     -- $S := S - \{s\}$
     -- $S := S \cup Min \{ h \mid h(x) = 1 \wedge \exists g \in G: g \supseteq h \supseteq s \}$
     -- $S := S - \{ h \mid \exists s \in S: h \supset s \}$

   Falls d negativ ($d = [x, 0]$):
   - $S := S - \{ s \mid s(x) \neq 0 \}$
   - Für alle $g \in G$ mit $g(x) \neq 0$:
     -- $G := G - \{g\}$
     -- $G := G \cup Max \{ h \mid h(x) = 0 \wedge \exists s \in S: g \supseteq h \supseteq s \}$
     -- $G := G - \{ h \mid \exists g \in G: g \supset h \}$
2. Ergebnis: $G = Max(VS_{H,D})$, $S = Min(VS_{H,D})$

Candidate-Elimination-Algorithmus

- Ergebnis korrekt, falls
  - Trainings-Beispiele korrekt
    andernfalls wird korrektes Konzept eliminiert
  - Zielkonzept in H enthalten
    sonst evtl. kein Ergebnis (alle Hypothesen verworfen)
- Dabei gilt:
  - Ergebnis enthält alle alternativen Lösungen
  - Abbruchmöglichkeit: falls Ergebnis eindeutig ($S = G = \{h\}$)

Weitere Probleme für Lernsysteme

(am Beispiel *Candidate-Elimination*-Algorithmus)

- Reihenfolge der Beispiele aus D:

Optimale Query-Strategie wählt Beispiel d aus, bei dem der aktuell konstruierte Version Space VS möglichst gleichmäßig aufgeteilt wird in:

$VS(d+) := \{ h \in V \mid h(d) = 1 \}$
$VS(d-) := \{ h \in V \mid h(d) = 0 \}$

allgemeines Prinzip: Entropie verringern

Weitere Probleme für Lernsysteme

(am Beispiel *Candidate-Elimination*-Algorithmus)

Anwendung partiell gelernter Konzepte C:

Sei H eine Hypothesenmenge mit $C \in H$.
Klassifikation ($x \in C$?) eines unbekannten Objekts x ist mittels H möglich, falls h(x) für alle $h \in H$ den gleichen Wert ergibt.

Es gilt: $x \in C$, falls $h(x)=1$ für alle $h \in Min\ H$ (speziell: S)
$x \notin C$, falls $h(x)=0$ für alle $h \in Max\ H$ (speziell: G)

Allgemeineres Prinzip: Voting
(gemäß Anzahl/Wahrsch. der Hypothesen $h \in H$ mit $h(x)= 1$ bzw. $=0$)

„Inductive Bias“

- Einschränkung des Hypothesenraums
  Konsequenz: Evtl. keine Lösung
- Vergrößerung des Hypothesenraums
  Konsequenz:
  - Verschlechterung der Generalisierungsfähigkeit
  - Extremfall: Alle Hypothesen
    Wenn beim Lernen mit korrekten Beispielen alle konsistenten Hypothesen behalten werden, gibt es keine Generalisierung (Voting für unbekannte Beispiele 50:50)

„Inductive Bias“

Generalisierung (Klassifizierung neuer Objekte) nur möglich,
wenn a priori Zusatzannahmen getroffen werden.
(Hier: Beschränkung des Hypothesenraum - „restriction bias“.)

- Zusatzannahme: **Inductive Bias**
- Beispiele:
  - Datenbank positiver/negativer Beispiele: kein bias
    - nur Beispiel-Objekte klassifizierbar
  - *Candidate Elimination*: Hypothesenraum enthält Zielkonzept
    - neue Objekte ggf. bei eindeutigem Votum klassifizierbar
  - *Find-S*: Hypothesenraum enthält Zielkonzept
    Ungesehene Objekte sind negativ (CWA)
    - alle Objekte klassifizierbar, solange Hypothese existiert

„Inductive Bias“

Generalisierung nur möglich,
wenn a priori Zusatzannahmen getroffen werden.

- Zusatzannahme: **Inductive Bias**

In der Praxis oft implizite Annahmen als inductive bias.

Analyse erforderlich

ZUSAMMENFASSUNG

- Konzeptlernen kann als Suchproblem spezifiziert werden.
- Algorithmus *Find-S* benutzt partielle Ordnung “*General-to-Specific*”. Entlang eines Pfades wird die spezifischste Hypothese gefunden, die mit den Trainingsdaten konsistent ist.
- Algorithmus *Candidate-Elimination* berechnet obere/untere Grenzen für Version Space. Der Version Space beschreibt vollständig alle verfügbaren Hypothesen der Zielfunktion (und ermöglicht damit Abbruchkriterium, Inkonsistenz-Prüfung der Trainingsdaten, optimale Auswahl von Trainingsbeispielen, Voting).
- *Find-S* und *Candidate-Elimination* sind nicht geeignet für fehlerhafte Trainingsdaten. Die Zielfunktion muss im Hypothesenraum enthalten sein.
- Generalisierung erfordert einen ***inductive bias***, d.h. a-priori gegebene Auswahlkriterien für Mengen konsistenter Hypothesen.