

Machine Learning Übungsblatt 3

Ramon Leiser Tobias Hahn

November 29, 2016

1 Begriffsdefinitionen

1.1 Hidden Markov Model - Bias

1.2 Inductive Learning Hypothesis

Die induktive Lernhypothese sagt dass eine Hypothese welche alle Trainingsdaten gut vorhersagen kann auch neue Testdaten gut vorhersagt. Das stimmt deswegen nicht, weil unbekannt ob sich neue Testdaten auch an die vorherigen halten. Nimmt man Beispiel an dass das Wetter für einen Tag bekannt ist und daraus bestimmt wird ob ich zur Uni gehe oder nicht, so könnten die Trainingsdaten folgendermaßen aussehen:

Wetter	Ich gehe zur Uni
Sonnig	Ja
Sonnig	Ja
Regen	Nein

Eine Hypothese welche die Daten gut beschreibt wäre {Sonnig}. Nun kann aber ein Tag kommen an dem ich zur Uni gehen muss obwohl es regnet - beispielsweise weil gerade Machine Learning Vorlesung ist und ich die natürlich nicht verpassen will. Dass würde der Hypothese widersprechen.

2 Concept Learning

2.1 Frage

Das Konzept bedeutet dass das Fortbewegungsmittel zwei Räder haben muss und per Pedal angetrieben wird, und eine beliebige Farbe bzw. Aussehen hat (jedoch nicht keine haben darf!). Wahrscheinlich handelt es sich darum um ein Fahrrad, da dieses zwei Räder hat und per Pedal angetrieben wird, während es immer irgendeine Farbe und irgendein Aussehen hat.

2.2 Trike

Ein Trike hat irgendeine Farbe und irgendein Aussehen, immer drei Räder und entweder einen Motor oder Pedale. Da es jedoch keine Ausdrucksweise dafür gibt dass ein Attribut nur gewisse Werte, andere jedoch nicht annehmen kann, so bleiben =zwei Alternativen übrig. Entweder es werden zwei Hypothesen angegeben ({Drei,Pedal,?,?} und {Drei,Motor,?,?}) oder wird eine allgemeine angegeben {Drei,?,?,?} worunter jedoch auch ein Trike mit Pferd Antrieb fallen würde.

2.3 Beispielmenge

Für jede mögliche Ausprägung eines Attributs gibt es jede mögliche Kombination der anderen Attribute. Wir müssen also die Anzahl der möglichen

Ausprägungen für jedes Attribut miteinander multiplizieren, und erhalten die Zahl $4 * 3 * 3 * 3 = 108$.

2.4 Mögliche Hypothesen mit Bias

Bei Hypothesen kann jedes Attribut entweder genau belegt werden, beliebig sein oder ausgeschlossen. Damit haben wir für die vorherige Rechnung bei jedem Attribut noch zwei weitere Ausprägungen hinzufügen, um die Anzahl der möglichen Hypothesen berechnen zu können: $6 * 5 * 5 * 5 = 750$. Das bedeutet dass nicht allzu viele Hypothesen zu überprüfen sind - der Algorithmus sollte nicht lange brauchen die optimale zu finden.

2.5 Mögliche Hypothesen ohne Bias

Ohne Bias können wir für eine Beobachtung nie sagen ob sie nun zu einer generelleren Aussage führt, deswegen können wir neue Beobachtungen immer nur mit den alten verunden. Dies führt zu einem Hypothesenraum von 2^{108} da jede mögliche Belegung der Attribute entweder in der Und-Kette vorhanden sein kann oder nicht. Das sind sehr sehr viele Hypothesen, ca. $3 * 10^{32}$.

2.6 Genereller-als

Die Hypothesen sind folgendermaßen geordnet (von der generellsten zur spezifischsten): $\{?, ?, ?, ?\}$, $\{\text{Eins}, ?, ?, ?\}$, $\{?, ?, ?, ?\}$, $\{?, \text{Pedal}, \text{Blau}, ?\}$

2.7 FIND-S

Auf die Trainingsdaten aus Tabelle 1 am Angabenblatt wird der Find S Algorithmus angewandt.

$$h_0 = \{, , , \}$$

2.7.1 1. Schritt

Datensatz: $\{\text{Zwei}, \text{Pedal}, \text{Grün}, \text{Wunderschön}\}$

Neue Hypothese $h_1 = \{\text{Zwei}, \text{Pedal}, \text{Grün}, \text{Wunderschön}\}$

2.7.2 2. Schritt

Datensatz: $\{\text{Zwei}, \text{Pedal}, \text{Blau}, \text{Wunderschön}\}$

Neue Hypothese $h_2 = \{\text{Zwei}, \text{Pedal}, ?, \text{Wunderschön}\}$

2.7.3 3. Schritt

Datensatz: $\{\text{Zwei}, \text{Motor}, \text{Rot}, \text{Wunderschön}\}$

Neue Hypothese $h_2 = \{\text{Zwei}, ?, ?, \text{Wunderschön}\}$

2.7.4 Ergebnis

Der Algorithmus kommt zu folgender Hypothese: $h = \{\text{Zwei}, ?, ?, \text{Wunderschön}\}$

Das ist wahrscheinlich nicht die beste Hypothese, da auch nicht wunderschöne Räder geeignet sind. Jedoch kommen im Datensatz nur wunderschöne Räder als Positivbeispiele vor, weswegen die Hypothese auch so aussieht. Außerdem kommt kein Positivbeispiel mit Pferd als Antrieb vor, dennoch sind dank Bias auch Pferde als Antrieb geeignet in unserer Hypothese.

2.8 Candidate Elimination Algorithmus

Der Candidate Elimination Algorithmus wird auf unsere Daten angewandt.

$$S_0 = \{, , , \}$$

$$G_0 = \{?, ?, ?, ?\}$$

2.8.1 1. Schritt

Datensatz = $\{Vier, Pferd, Rot, schön\}$

Vorgehen Da der Datensatz ein Negativbeispiel ist müssen zuerst alle Hypothesen aus S entfernt werden die konsistent mit dem Negativbeispiel sind, die es also nicht ausschließen. Das trifft auf die einzige Hypothese in S nicht zu, daher verändert sich S nicht.

Aus G müssen alle Hypothesen entfernt werden die nicht konsistent mit dem Negativbeispiel sind, die es also beschreiben. Dies trifft auf die einzige Hypothese in G zu, deshalb wird sie entfernt. Alle minimalen Spezifikationen (jeweils eines der Attribute mit einer der Ausprägungen, ohne die im Beispiel) sind konsistent mit der einzigen Hypothese in S, weswegen alle hinzugefügt werden.

Neue Hypothesenmengen

$$S_0 = \{, , , \}$$

$$G_0 = \{\{Eins, ?, ?, ?\}, \{Zwei, ?, ?, ?\}, \{Drei, ?, ?, ?\},$$

$$\{?, Pedal, ?, ?\}, \{?, Motor, ?, ?\}, \{?, ?, Grün, ?\},$$

$$\{?, ?, Blau, ?\}, \{?, ?, ?, hässlich\}, \{?, ?, ?, wunderschön\}\}$$

2.8.2 2. Schritt

Datensatz = $\{Zwei, Pedal, Grün, wunderschön\}$

Vorgehen Da der Datensatz ein Positivbeispiel ist müssen zuerst alle Hypothesen aus G entfernt werden die nicht konsistent mit dem Positivbeispiel sind, die es also ausschließen. Das trifft auf alle Hypothesen zu die ein Attribut mit einem anderen Wert als unser Beispiel ausgeprägt haben.

Aus S müssen wiederum alle Hypothesen entfernt werden die nicht konsistent mit dem Beispiel sind, die es also ausschließen. Das trifft auf die spezifischste Hypothese zu, sie wird also entfernt. Nun müssen noch die geringsten Generalisierungen der entfernten These hinzugefügt werden, die konsistent mit dem Datensatz sind und für die eine These aus G genereller ist. Dies trifft auf den Datensatz als These zu.

Neue Hypothesenmengen

$$S_0 = \{Zwei, Pedal, Grün, wunderschön\}$$

$$G_0 = \{\{Zwei, ?, ?, ?\}, \{?, Pedal, ?, ?\}, \{?, ?, Grün, ?\}, \{?, ?, ?, wunderschön\}\}$$

2.8.3 3. Schritt

$$\text{Datensatz} = \{Vier, Motor, Grün, hässlich\}$$

Vorgehen Da der Datensatz ein Negativbeispiel ist müssen zuerst alle Hypothesen aus S entfernt werden die konsistent mit dem Negativbeispiel sind, die es also nicht ausschließen. Das trifft auf die einzige Hypothese in S nicht zu, daher verändert sich S nicht.

Aus G müssen alle Hypothesen entfernt werden die nicht konsistent mit dem Negativbeispiel sind, die es also beschreiben. Dies trifft auf die Hypothese mit Grün als Ausprägung für Farbe zu, diese wird entfernt. Nun werden noch alle Hypothesen hinzugefügt die eine minimale Spezialisierung der entfernten sind, das Beispiel ausschließen und genereller als die Hypothese aus S sind. Da drei Hypothesen Kollegen in G haben, die genereller sind werden sie entfernt.

Neue Hypothesenmengen

$$S_0 = \{Zwei, Pedal, Grün, wunderschön\}$$

$$G_0 = \{\{Zwei, ?, ?, ?\}, \{?, Pedal, ?, ?\}, \{?, ?, ?, wunderschön\}\}$$

2.8.4 4. Schritt

$$\text{Datensatz} = \{Eins, Motor, Grün, schön\}$$

Vorgehen Da der Datensatz ein Negativbeispiel ist müssen zuerst alle Hypothesen aus S entfernt werden die konsistent mit dem Negativbeispiel sind, die es also nicht ausschließen. Das trifft auf die einzige Hypothese in S nicht zu, daher verändert sich S nicht.

Aus G müssen alle Hypothesen entfernt werden die nicht konsistent mit dem Negativbeispiel sind, die es also beschreiben. Dies trifft auf keine Hypothese aus G zu, daher verändert sich G nicht.

Neue Hypothesenmengen

$$S_0 = \{Zwei, Pedal, Grün, wunderschön\}$$

$$G_0 = \{\{Zwei, ?, ?, ?\}, \{?, Pedal, ?, ?\}, \{?, ?, ?, wunderschön\}\}$$

2.8.5 5. Schritt

Datensatz = $\{Zwei, Pedal, Blau, wunderschön\}$

Vorgehen Da der Datensatz ein Positivbeispiel ist müssen zuerst alle Hypothesen aus G entfernt werden die nicht konsistent mit dem Positivbeispiel sind, die es also ausschließen. Das trifft auf keine der Hypothesen zu, G verändert sich also nicht.

Aus S müssen wiederum alle Hypothesen entfernt werden die nicht konsistent mit dem Beispiel sind, die es also ausschließen. Das trifft auf die spezifischste Hypothese zu, sie wird also entfernt. Nun müssen noch die geringsten Generalisierungen der entfernten These hinzugefügt werden, die konsistent mit dem Datensatz sind und für die eine These aus G genereller ist. Dies trifft auf die These zu, welche die Farbe beliebig stellt.

Neue Hypothesenmengen

$$S_0 = \{Zwei, Pedal, ?, wunderschön\}$$
$$G_0 = \{\{Zwei, ?, ?, ?\}, \{?, Pedal, ?, ?\}, \{?, ?, ?, wunderschön\}\}$$

2.8.6 6. Schritt

Datensatz = $\{Zwei, Motor, Rot, wunderschön\}$

Vorgehen Da der Datensatz ein Positivbeispiel ist müssen zuerst alle Hypothesen aus G entfernt werden die nicht konsistent mit dem Positivbeispiel sind, die es also ausschließen. Das trifft auf die Hypothese mit der Ausprägung Pedal zu, diese wird also entfernt.

Aus S müssen wiederum alle Hypothesen entfernt werden die nicht konsistent mit dem Beispiel sind, die es also ausschließen. Das trifft auf die spezifischste Hypothese zu, sie wird also entfernt. Nun müssen noch die geringsten Generalisierungen der entfernten These hinzugefügt werden, die konsistent mit dem Datensatz sind und für die eine These aus G genereller ist. Dies trifft auf die These zu, welche die Antriebsart beliebig stellt.

Neue Hypothesenmengen

$$S_0 = \{Zwei, ?, ?, wunderschön\}$$
$$G_0 = \{\{Zwei, ?, ?, ?\}, \{?, ?, ?, wunderschön\}\}$$

Deutung Dieses Ergebnis ergibt, dass Fahrradkuriere zweirädrige, wunderschöne Fortbewegungsmittel bevorzugen.

2.9 Naive Bayes

Der Gelernte Baum sieht folgendermaßen aus:

Root: Classifier

Base probabilities:

decisionTreeC45: 0.07692308, kNearestNeighbour: 0.07692308, naiveBayes:
0.07692308, decisionTreeID3: 0.07692308, candidateElimination: 0.07692308,

Nodes:

Trainingsetsize

5 x 2 big

decisionTreeC45 | low: 0.0

decisionTreeC45 | high: 1.0

kNearestNeighbour | low: 1.0

kNearestNeighbour | high: 0.0

naiveBayes | low: 0.33333334

naiveBayes | high: 0.6666667

decisionTreeID3 | low: 0.6666667

decisionTreeID3 | high: 0.33333334

candidateElimination | low: 0.6666667

candidateElimination | high: 0.33333334

Domain

5 x 2 big

decisionTreeC45 | discrete: 0.0

decisionTreeC45 | continuous: 1.0

kNearestNeighbour | discrete: 0.33333334

kNearestNeighbour | continuous: 0.6666667

naiveBayes | discrete: 0.0

naiveBayes | continuous: 1.0

decisionTreeID3 | discrete: 1.0

decisionTreeID3 | continuous: 0.0

candidateElimination | discrete: 1.0

candidateElimination | continuous: 0.0

Noise

5 x 3 big

decisionTreeC45 | none: 1.0

decisionTreeC45 | high: 0.0

decisionTreeC45 | low: 0.0

kNearestNeighbour | none: 0.0

kNearestNeighbour | high: 0.33333334

kNearestNeighbour | low: 0.6666667

naiveBayes | none: 0.0

naiveBayes | high: 1.0

naiveBayes | low: 0.0

decisionTreeID3 | none: 1.0

decisionTreeID3 | high: 0.0

decisionTreeID3 | low: 0.0

candidateElimination | none: 1.0

```
candidateElimination | high: 0.0
candidateElimination | low: 0.0
```

Die Wahrscheinlichkeiten der Zugehörigkeit für die Trainingsdaten zu den jeweiligen Klassifikationsgruppen sieht folgendermaßen aus:

```
(Domain) (Trainingsetsize) (Noise)
continious, high, high -> decisionTreeC45 with 0.0
continious, high, high -> kNearestNeighbour with 0.0
continious, high, high -> naiveBayes with 0.15384616
continious, high, high -> decisionTreeID3 with 0.0
continious, high, high -> candidateElimination with 0.0
```

```
discrete, high, none -> decisionTreeC45 with 0.0
discrete, high, none -> kNearestNeighbour with 0.0
discrete, high, none -> naiveBayes with 0.0
discrete, high, none -> decisionTreeID3 with 0.07692308
discrete, high, none -> candidateElimination with 0.07692308
```

```
continious, high, low -> decisionTreeC45 with 0.0
continious, high, low -> kNearestNeighbour with 0.0
continious, high, low -> naiveBayes with 0.0
continious, high, low -> decisionTreeID3 with 0.0
continious, high, low -> candidateElimination with 0.0
```

Auffällig ist dass das letzte Beispiel nicht klassifiziert wird weil die Multiplikation der einzelnen Wahrscheinlichkeiten zusammen immer null ergibt. Dieses Problem des Naive Bayes wurde auch auf den Folien zum Klassifikator auf Seite 42 genannt. Als mögliche Lösung wird vorgeschlagen nur Wahrscheinlichkeiten über Eins zu benutzen. Dies erzeugt zwar mathematisch falsche Wahrscheinlichkeitswerte, erlaubt aber manchmal eine Klassifikation wo konventionell kein Ergebniss erreicht wird.

Auf dem letzten Eintrag der Testdaten ergibt sich dann:

```
continious, high, low -> decisionTreeC45 with 4.3076925
continious, high, low -> kNearestNeighbour with 3.4188037
continious, high, low -> naiveBayes with 4.1025643
continious, high, low -> decisionTreeID3 with 1.6410258
continious, high, low -> candidateElimination with 1.6410258
```

Man erkennt dass der Testfall bei dieser Rechnung besser zugeordnet werden kann und sich ein starker Unterschied der Werte zwischen den ersten drei Algorithmen und den letzten zwei erkennen lässt.