27 Klassifikation

27.1 (Boolesche) Entscheidungsfunktionen

Klassifikation ist ein erstes Beispiel für induktive Inferenz. Was dabei induziert wird ist eine Funktion f, und zwar eine diskrete Funktion, d.h. eine Funktion die nur endlich viele verschiedene Eingaben nimmt und damit nur endliche viele Ausgaben liefert. Wir werden uns hauptsächlich einen Spezialfall der Klassifikation anschauen, nämlich die **Boolesche Klassifikation**. Boolesche Klassifikation ist deswegen speziell, weil wir eine Boolesche (Wahrheits-)Funktion lernen. Wir suche eine Funktion,

- die für eine Eingabe x entweder "ja" oder "nein" liefert;
- wir fassen "ja" als 1, nein als 0 auf;
- weiterhin basiert eine solche Funktion auf einer Menge von **Attributen**, die auch entweder den Wert 0 oder 1 haben (das werden wir lockern), also erfüllt sind oder nicht.

Wir haben also eine Funktion

(405) $f : \{0,1\}^n \to \{0,1\}$

Um mit dem Konzept vertraut zu werden, erstmal folgendes Beispiel (aus Russel & Norvig): es geht um die Entscheidung, ob wir in einem Restaurant warten, bis wir einen Tisch zugewiesen bekommen, oder weitergehen; also eine binäre Entscheidung. NB: wir suchen also unsere eigene Entscheidungsfunktion, möchten also eine Funktion die uns für jedes Restaurant sagt, ob wir warten würden!

Die Attribute sind hier nicht alle binär, aber das tut erstmal nichts zur Sache. Als erstes stellen wir die Liste der Merkmale zusammen, die für unsere Entscheidungsfunktion relevant sind (schöner wäre es natürlich, wenn wir diese Attribute automatisch erstellen könnten, dazu später mehr). Unsere Merkmale sind:

1. Alternativen: gibt es passende Alternativen in der Nähe?
2. Theke: können wir uns an die Theke setzen und schonmal ein Bier trinken?
3. Fr/Sa: ist es Freitag oder Samstag?
4. Betrieb: wie viel Betrieb ist im Lokal? (Werte: leer, einige Leute, voll)
5. Regen: regnet es draußen?
6. Reservierung: haben wir reserviert?
7. Typ: was für eine Art Restaurant haben wir (französisch, italienisch, deutsch)
8. Geschätzte Wartezeit (von uns geschätzt): 0-10,10-30,30-60,>60

Das sind also die Faktoren, die bestimmen, ob wir auf einen freien Tisch warten. Nicht alle Attribute sind binär; wie können sie aber leicht darauf reduzieren; z.B. Attribut 4. kann aufgespalten werden in 2 Attribute: Leer: ja/nein und Voll: ja/nein. Unser Hypothesenraum besteht also aus allen Funktionen

(406) $h : \{0,1\}^3 \times \{0,1,2\} \times \{0,1\}^2 \times \{0,1,2\} \times \{0,1,2,3\} \to \{0,1\}$

Wie viele solche Funktionen gibt es? Nehmen wir einfachheitshalber mal an, **H** wäre die Menge aller Funktionen

(407) $h' : \{0,1\}^8 \to \{0,1\}$

Wie groß ist unser Hypothesenraum? Man könnte meinen er wäre nicht übermäßig groß; aber der Eindruck täuscht:

$$\text{es gibt } 2^{2^8} \text{ solche Funktionen, also } 2^{64}$$

– eine wahnsinnig große Zahl. Unser Hypothesenraum ist also riesig! Unser Ziel muss es sein, eine möglichst einfache Funktion aus diesem Raum zu wählen, die (nach unseren Begriffen) gut verallgemeinert. Hierbei greift man auf die sogenannten **Entscheidungsbäume** zurück.

Bsp.	Alt	Theke	Fr/Sa	Bet	Reg	Res	Typ	Wart	Warten?
d1	1	0	0	halb	0	1	fr	0-10	1
d2	1	0	0	voll	0	0	it	30-60	0
d3	0	1	0	halb	0	0	de	0-10	1
d4	1	0	1	voll	1	0	it	10-30	1
d5	1	0	1	voll	0	1	fr	>60	0
...									

Table 3: Ein Ausschnitt aus unserem Datensatz

27.2 Entscheidungsbäume

Boolesche Funktionen lassen sich einfach als Tabellen auffassen; wir nehmen nun wieder unser Beispiel, um das darzustellen: Tabelle 1 ist nur ein kleiner Ausschnitt unserer Funktion; wir können auch annehmen, es handelt sich um unseren Datensatz D. Ein **Entscheidungsbaum** ist einfach ein Baum,

1. in dem jeder Knoten ein Merkmal repräsentiert,
2. jedes Blatt einen Wert, den die Funktion annimmt;
3. auf jedem Pfad von der Wurzel zu einem Blatt kommt dabei jedes Merkmal höchstens einmal vor.

Jede Boolesche Funktion lässt sich als als Entscheidungsbaum darstellen: wir können einfach den Baum nehmen, in dem jede *Zeile* unserer Tabelle einem *Pfad* entspricht. Es gibt gewisse Boolesche Funktionen, die lassen sich nicht oder nur sehr schwer kompakt repräsentieren, z.B.

> die **Paritätsfunktion** (f nimmt den Wert 1 an, wenn eine gerade Zahl von Argumenten den Wert 1 annimmt), oder
>
> die **Majoritätsfunktion** (f nimmt den Wert 1 an, falls mindestens die Hälfte seiner Argumente den Wert 1 annimmt).

Allerdings gibt es auch Entscheidungsbäume, die eine wesentlich kompaktere Darstellung erlauben. Wenn wir das obige Beispiel betrachten, dann fällt uns z.B. auf dass wann immer die geschätzte Wartezeit >60 Minuten beträgt, dann warten wir niemals darauf dass ein Tisch frei wird. Wenn sich dieses Muster durch alle unsere Beobachtungen zieht, dann können wir also

dieses Merkmal an die Wurzel unseres Baumes setzen, und dann können wir in einigen Fällen den Baum an dieser Stelle schon mit dem Blatt 0 beenden. Algorithmen zur Induktion von Entscheidungsbäumen beruhen genau auf dieser Beobachtung:

> Wir können die Komplexität von Booleschen Funktionen messen nach der Komplexität der Entscheidungsbäume.

Das wiederum passt zu unseren obigen Beobachtung, dass einfache Funktionen eher sinnvolle, interessante Generalisierungen liefern als komplexe. Wir bekommen also folgendes:

> Gegeben eine Menge D von Daten, finde den enfachsten Entscheidungsbaum, der mit D konsistent ist; die zugehörige Boolesche Funktion ist unsere Hypothese h.

Wie finden wir die? Man benutzt hier das sog. **Splitting**: wir nehmen das Merkmal, dass für unsere Unterscheidung **am informativsten** ist, und setzen es an die Wurzel des Entscheidungsbaumes. Dann nehmen wir das nächst-informativste Merkmal, setzen es als nächsten Knoten etc. Wie macht man das? Hier nutzen wir wieder einmal das Konzept der **Entropie**. Dafür müssen wir zunächst etwas arbeiten:

- Unser zugrundeliegende Raum ist eine Menge von Funktionen $X : M_1 \times M_2 \times \ldots \times M_i \to \{0, 1\}$.
- Wenn wir nun ein $n : 1 \leq n \leq i$ wählen, dann haben wir eine Funktion $X_n : M_n \mapsto (M_1 \times \ldots \times M_{n-1} \times M_{n+1} \times \ldots \times M_i \to \{0, 1\})$
- Das bedeutet: für jedes Merkmal, dass einen gewissen Wert annimmt, bekommen wir eine neue Funktion über die verbliebenen Merkmale.
- Wir möchten das Merkmal finden, das uns am besten die Menge der verbliebenen Funktionen aufteilt; insbesondere sollten die Teilmengen disjunkt sein!

Wir suchen also erstmal Merkmale M_n, für die gilt:

> Falls $m, m' \in M_n$, $m \neq m'$, dann ist $X_n(m) \cap X_n(m') = \emptyset$.

Das ist aber ein Kriterium, das gleichzeitig zu schwach (viele Merkmale können es erfüllen) und zu stark ist (in manchen Fällen wird es kein Merkmal geben, dass dieses Kriterium erfüllt.

Wir müssen also mal wieder Zuflucht zu Wahrscheinlichkeiten nehmen. Wir bauen daher den Wahrscheinlichkeitsraum $\mathfrak{A}$, wobei gilt:

1. $\Omega = M_1 \times M_2 \times \ldots \times M_i \rightarrow \{0, 1\}$ (die Menge der Ereignisse),
2. und für jedes $d \in \Omega$ gilt:

$$P(d) = \frac{1}{|D|} \text{ falls } f \in D, \text{ wobei } D \text{ unser Datensatz ist.}$$

Auf diesem Raum können wir nun eine Reihe von Zufallsvariablen $X_n : n \leq i$ definieren (wir fassen hier den Begriff etwas allgemeiner):

$$\text{Für } d = (m_1, \ldots, m_n, \ldots, m_i, x) \ (x \in \{0, 1\}),$$

gilt:

$$X_n(d) = m_n.$$

Man beachte, dass der **Zielwert** x (0 oder 1) hier nur ein weiteres Merkmal unter vielen ist! Nun hat jede dieser Zufallsvariablen eine Entropie, die sich errechnet als

$$(408) \quad H_P(X_n) = \sum_{m \in M_n} P(X_n = m) \cdot log(P(X_n = m))$$

Damit bemessen wir, wie informativ eine Variable ist, und da die Variablen einem Merkmal entsprechen, bemessen wir also indirekt, wie informativ ein Merkmal ist. Das wäre aber zu allgemein: wir möchten ja nicht irgendein Merkmal vorhersagen, sondern ein ganz bestimmtes, unser **Zielmerkmal**. Hierzu brauchen wir das Konzept der **bedingten Entropie**:

$$\begin{aligned} H(X|Y) &= \sum_{y \in Y} H(X|Y = y) \\ &= \sum_{x \in X, y \in Y} P(X^{-1}(x) \cap Y^{-1}(y)) log \left(\frac{P(X^{-1}(x) \cap Y^{-1}(y))}{P(Y^{-1}(y))} \right) \end{aligned}$$

Insbesondere interessiert uns die Entropie der Variable X_{Ziel}, also des Zielwertes, gegeben dass wir den Wert eines Merkmals kennen:

$$(409) \quad H_P(X_{Ziel}|X_n)$$

Was jedoch wichtiger ist als dieser Wert (der ja auch sehr extrem sein kann, auch wenn M_n keinen Einfluss darauf hat) ist der **Informationsgewinn**; der ist wie folgt definiert:

(410) $IG_P(X_{Ziel}|X_n) = H_P(X_{Ziel}) - H_P(X_{Ziel}|X_n)$

Je geringer die bedingte Entropie im Vergleich zur unbedingten ist, desto größer ist der Informationsgewinn. Falls

(411) $H_P(X_{Ziel}|X_n) = 0$

also der Wert von X_{Ziel} vollständig von X_n bestimmt wird, dann ist

(412) $IG_P(X_{Ziel}|X_n) = H_P(X_{Ziel})$

Das bedeutet: wir gewinnen sämtliche Information, die in X_{Ziel} enthalten ist. Was wir damit also suchen ist:

(413) $\underset{1 \leq n \leq i}{argmax}\ IG_P(X_{Ziel}|X_n)$

Das liefert uns das Merkmal, welches wir ganz oben in unseren Entscheidungsbaum stellen. Danach iterieren wir das mit den verbliebenen Variablen/Merkmalen: als nächstes interessiert uns

(414) $\underset{1 \leq n \leq i}{argmax}\ H_P(X_{Ziel}|X_{max}) - H_P(X_{Ziel}|X_{max}, X_n)$

und so weiter, so dass wir also $i!$ Schritte benötigen (ein Schritt ist hier die Berechnung der bedingten Entropie). Das ist ein gutes Ergebnis, da die Anzahl der Merkmale normalerweise überschaubar ist!

27.3 Overfitting I

Vorher haben wir die Tatsache benutzt, das gewisse Merkmale informativer sind als andere. Es gibt hierbei aber ein mögliches Problem: dass ein Merkmal *zu informativ* ist, nämlich keine Generalisierung enthält. Das passiert insbesondere, wenn das Merkmal viele Werte annehmen kann, schlimmstenfalls mehr als unser Datensatz an Punkten enthält. Ein Beispiel hierfür wäre, wenn wir ein Merkmal **Datum** hinzunehmen. Unter der Annahme, dass wir an jedem Tag nur einmal essen gehen, ist klar dass wir damit einen perfekten

Bsp.	Alt	Theke	Fr/Sa	Bet	Reg	Res	Typ	Wart	Tag	Warten?
d1	1	0	0	halb	0	1	fr	0-10	5	1
d2	1	0	0	voll	0	0	it	30-60	18	0
d3	0	1	0	halb	0	0	de	0-10	9	1
d4	1	0	1	voll	1	0	it	10-30	26	1
d5	1	0	1	voll	0	1	fr	>60	17	0
...										

Table 4: Die Daten mit dem Tag des Monats

Prädiktor für unseren Datensatz haben: das Datum gibt uns eindeutig die richtige Klassifizierung. Das Problem ist: es gibt dabei keine Generalisierung! Das bleibt bestehen wenn wir ein Merkmal haben **Tag des Monats** – auch das mag bei einem relativ kleinen Datensatz ein guter Prädiktor sein, hat aber vermutlich keine Relevanz.

Das zugrundeliegende Problem ist also, dass

$$\frac{|M|}{|D|},$$

der relativ groß ist, im schlimmsten Fall > 1. Wie gehen wir mit diesem Merkmal um? Wir können ja nicht davon ausgehen, dass die Irrelevanz eines Merkmals derart offen zutage liegt. Hier können wir die klassische statistische Analyse nutzen: die **Nullhypothese** ist, dass das Merkmal keinen Einfluss hatte auf unsere jeweilige Entscheidung. Wir können nun versuchen, diese Hypothese zu widerlegen: wir müssen belegen, dass es wahrscheinlich ist, dass die Verteilung des Merkmals M rein zufällig ist.

Dafür überlegen wir zunächst:

- Wie viele Werte kann das Merkmal M annehmen? Wir nennen diese Zahl $|M|$.
- Wie würde es aussehen, wenn diese Merkmale rein zufällig über die anderen verteilt würden? Es würde zunächst gleichmäßig gestreut sein, d.h. keine besondere Ko-Okkurenz mit anderen Merkmalen haben.

Den zweiten Punkt kann man wie folgt verdeutlichen: da $|M|$ in kritischen Fall relativ groß ist. muss man ein Merkmal M' nehmen mit möglichst kleinem $|M'|$. Ein besonderes Beispiel hierfür wäre das "Zielmerkmal" $\{0, 1\}$, das wir eigentlich vorhersagen möchten. In diesem Fall ist die **Nullhypothese** klar numerisch formulierbar; wir benutzen unsere Zufallsvariablen X_n, setzen fest (qua Definition):

$$M = M_j \qquad M' = M_k$$

Nun sollte laut Nullhypothese gelten:

für alle $m \in M_j, m' \in M_k$, $d \in D$:
$$P(X_j = m | X_k = m') \approx P(X_j = m)$$

Da $P(X_j = m)$ aber naturgemäss (qua Annahme dass $\frac{|M|}{|D|}$ relativ groß ist) eine Zahl ist, die für uns schwierig von 0 zu unterscheiden ist, ist das noch problematisch; wir können aber folgendes machen: nehmen wir Einfachheit halber an, alle anderen Merkmale außer M sind binär. Dann können wir eine neue Zufallsvariable Y annehmen, die eine Summe von Werten denotiert:

$$(415)\ Y(M) = \sum_{j \neq k} P(X_j = m | X_k = m')$$

(wobei i die Gesamtanzahl der Merkmale ist) als das Ergebnis eines $i-1$-Fach wiederholten Zufallsexperimentes lesen, wobei jeweils mit einem sehr großen Würfel geworfen wurde. Dementsprechend haben wir also eine Multinomialverteilung mit einem Erwartungswert

$$(416)\ \mathcal{E}(Y) = \frac{i-1}{|M|}$$

mit einer entsprechenden symmetrischen Verteilung, Varianz und Standardabweichung. Das bedeutet: wir können die üblichen Methoden der Vertrauensgrenzen etc. ohne weiteres anwenden.

27.4 Overfitting II

Wir können auch im Rahmen unserer Methodik der Informationstheorie bleiben, und den Begriff der **bedingten Entropie** nutzen. Hier nochmals die Definition:

$$\begin{aligned} H(X|Y) &= \sum_{y \in Y} H(X|Y = y) \\ &= \sum_{x \in X, y \in Y} P(X^{-1}(x) \cap Y^{-1}(y)) log \left(\frac{P(X^{-1}(x) \cap Y^{-1}(y))}{P(Y^{-1}(y))} \right) \end{aligned}$$

Nach unseren Annahmen für $M = M_j$ hat die Zufallsvariable X_j sicher eine hohe/maximale Entropie. Es ist also genau die Eigenschaft, die sie eigentlich

positive hervorheben, die sie auch problematisch macht! Hier sehen wir die zwei Seiten derselben Medaille: je größer $|M_j|$, desto größer die Entropie von $H_P(X_j)$; aber je größer $|M_j|$, desto größer die Gefahr, dass das Merkmal eigentlich keine relevante Information enthält. Wir können uns nun mit der bedingten Entropie helfen: sei X_{Ziel} die Zufallsvariable, die das Zielmerkmal unserer Daten liefert. Wir können nun z.B.

(417) $H_P(X_j|X_{Ziel})$

berechnent. Falls nun gilt:

(418) $H_P(X_j|X_{Ziel}) \approx H_P(X_j)$

dann wissen wir, dass das Ergebnis einen geringen Einfluss auf X_j hat (den Tag des Monats). Im Umkehrschluss bedeutet dass, dass auch andersrum wenig Information fließt; wir haben zwar diesen Eindruck, aber das ist nur der Größe $|M_j|$ geschuldet.