Seminar

Rough Sets

Präsentation von Christian Köllner

Betreuer: Philipp Bender

Wintersemester 2003 / 04

Rough set theory

- In den frühen 1980ern von **Zdislaw Pawlak** eingeführt
- Ein mathematisches Instrument, um mit **Unbestimmtheit** und **Unsicherheit** umzugehen
- Anwendungen in Bereichen der **künstlichen Intelligenz**, des **maschinellen Lernens**, der **Entscheidungsanalyse**, der **Wissensaneignung von Datenbanken**, **Mustererkennung** und vielen weiteren

Grundbegriff: Informationssystem

	Attribute			*Entscheidung*
	Kopf-schmerzen	Muskel-schmerzen	Temperatur	Grippe
e1	ja	ja	normal	nein
e2	ja	ja	hoch	ja
e3	ja	ja	sehr hoch	ja
e4	nein	ja	normal	nein
e5	nein	nein	hoch	nein
e6	nein	ja	sehr hoch	ja

Grundbegriff: Informationssystem

	Attribute			*Entscheidung*
	Kopf-schmerzen	Muskel-schmerzen	Temperatur	Grippe
e1	ja	ja	normal	nein
e2	ja	ja	hoch	ja
e3	ja	ja	sehr hoch	ja
e4	nein	ja	normal	nein
e5	nein	nein	hoch	nein
e6	nein	ja	sehr hoch	ja

```
Temperatur( e3 ) = "sehr hoch"
```

Ein Informationssystem ist:

- Ein Paar $S = (U, A)$
- U: Menge von Objekten, das **Universum**
- A: Menge von **Attributen** mit

$$a : U \rightarrow V_a \;\; \textit{für alle } a \in A$$

Grundbegriff: Informationssystem

	Attribute			*Entscheidung*
	Kopf-schmerzen	Muskel-schmerzen	Temperatur	Grippe
e1	ja	ja	normal	nein
e2	ja	ja	hoch	ja
e3	ja	ja	sehr hoch	ja
e4	nein	ja	normal	nein
e5	nein	nein	hoch	nein
e6	nein	ja	sehr hoch	ja

Ein Informationssystem ist:

- Ein Paar $S = (U, A)$
- U: Menge von Objekten, das **Universum**
- A: Menge von **Attributen** mit

$$a : U \rightarrow V_a \quad \text{für alle } a \in A$$

- Die Menge V_a heißt **Wertemenge** von a.

$$V_{Temperatur} = \{\texttt{"normal", "hoch", "sehr hoch"}\}$$

Grundbegriff: Entscheidungssystem

- Klassifizierung des Universums anhand eines bestimmten Attributes
- Ein **Entscheidungssystem** S hat die Form:

$$S = \left(U,\, A \cup \{d\}\right),\, d \notin A$$

- d heißt **Entscheidungsattribut**
- In unserem Fall: "**Grippe**"
- Kann beliebige Trägermenge haben
- Meistens jedoch binäre Ausprägung

Minimierung des Systems

- Entscheidungssystem beinhaltet komplettes Wissen über ein Modell
- Tabelle kann aber unnötig groß sein:
  - Mehrfach gleiche Zeilen
  - Überflüssige Attribute (dazu mehr gegen Ende der Präsentation)

⇨ Minimalität erwünscht

Ununterscheidbarkeit

	Attribute			*Entscheidung*
	Kopf-	Muskel-schmerzen	Temperatur	Grippe
e1	ja	ja	hoch	ja
e2	ja	ja	hoch	ja
e3	ja	ja	sehr hoch	ja
e4	ja	ja	sehr hoch	nein
e5	nein	nein	hoch	nein
e6	nein	nein	normal	ja

ununterscheidbar

widersprüchlich

intuitiv widersprüchlich

Ununterscheidbarkeit

- Sei $S = (U, A)$ ein Informationssystem
- Für jedes $B \subseteq A$ ist die **Äquivalenzrelation** $IND_A(B)$ definiert durch:

$$\boxed{IND_A(B) = \left\{ \left(x, x'\right) \in U^2 \,\middle|\, \forall a \in B \; a(x) = a(x') \right\}}$$

- Alle Elemente sind äquivalent, die bezüglich ihrer Attribute in B nicht unterscheidbar (d.h. gleich) sind.
- $IND_A(B)$ heißt **B-Ununterscheidbarkeitsrelation**

Ununterscheidbarkeit

	Attribute			*Entscheidung*
	Kopf-schmerzen	Muskel-schmerzen	Temperatur	Grippe
e1	ja	ja	normal	nein
e2	ja	ja	hoch	ja
e3	ja	ja	sehr hoch	ja
e4	nein	ja	normal	nein
e5	nein	nein	hoch	nein
e6	nein	ja	sehr hoch	ja

$$IND(\{\text{Grippe}\}) = \{\ \{e1, e4, e5\}, \{e2, e3, e6\}\ \}$$

Rough Sets

- Die Äquivalenzrelation induziert eine Partition des Universums.
- Partitionen definieren bestimmte Teilmengen des Universums.
- Die Teilmengen, die im Entscheidungsattribut den gleichen Wert haben, sind von Interesse.
- Ein Konzept wie z.B. "Grippe" kann jedoch aus gegebenen Daten nicht immer scharf definiert werden.
- Man benötigt unscharfe Mengen, die **Rough Sets**.

Rough Sets

	Kopf-schm.	Muskel-schm.	Temp.	Grippe
e1	ja	ja	normal	nein
e2	ja	ja	hoch	ja
e3	ja	ja	sehr hoch	ja
e4	nein	ja	normal	nein
e5	nein	nein	hoch	nein
e6	ja	ja	normal	ja

Klassifizierung in…

- sicherlich gesunde Patienten (e4, e5)
- sicherlich kranke Patienten (e2, e3)
- Grenzfälle (e1, e6).

- **Ist die Menge der Grenzfälle nicht leer, so ist die zu klassifizierende Menge unscharf (*engl.: rough*)**

Mengenapproximation

- Sei $\mathcal{A} = (U, A)$ ein Informationssystem und sei $B \subseteq A$ und $X \subseteq U$.

- Dann heißt $\underline{B}X = \left\{ x \mid [x]_B \subseteq X \right\}$ **untere B-Approximation von X**.
- Und $\overline{B}X = \left\{ x \mid [x]_B \cap X \neq \phi \right\}$ heißt **obere B-Approximation von X**.

Mengenapproximation

$$\underline{B}X = \{\, x \mid [x]_B \subseteq X \,\} \qquad \overline{B}X = \{\, x \mid [x]_B \cap X \neq \phi \,\}$$

- Elemente von $\underline{B}X$ sind mit Sicherheit in X.
- Elemente von $\overline{B}X$ sind nur eventuell in X.
- Die Menge $BN_B(X) = \overline{B}X - \underline{B}X$ heißt **B-Grenzregion** von X.
- Die Menge $U - \overline{B}X$ heißt **B-Außenregion** von X. Sie enthält Elemente, die sicher nicht zu X gehören.

Mengenapproximation

x	*Temperatur*	*krank*
x1	37 - 38	Nein
x2	38 - 39	Ja
x3	38 - 39	Nein
x4	> 39	Ja

$$X := \{\, x \mid krank(x) = Nein \,\}$$

$$= \{\, x1, x3 \,\}$$

$$B := \{\, Temperatur \,\}$$

$$U - \overline{B}X = \{x4\}$$

Ja

$$\underline{B}X = \{x1\}$$

Nein

$$BN_B(X) = \{x2, x3\}$$

Ja/Nein

Ermittlung von Redukten

- Problem kann auf einen Algorithmus zur **Minimierung boolescher Ausdrücke** reduziert werden (darum auch NP-hart!)
- Ausgangspunkt ist die **Unterscheidbarkeitsmatrix** $((c_{ij}))$

$$c_{ij} = \left\{ a \in A \,\middle|\, a(x_i) \neq a(x_j) \right\} \quad i, j = 1, \ldots, n$$

- **Interpretation:** c_{ij} = { *Menge der Attribute, in denen sich* x_i *von* x_j *unterscheidet* }

Unterscheidbarkeitsmatrix

x	*Temperatur* **[a]**	*krank* **[b]**
x1	37 - 38	Nein
x2	38 - 39	Ja
x3	38 - 39	Nein
x4	> 39	Ja

Informationssystem

	x1	x2	x3	x4
x1	Ø			
x2	{ a, b }	Ø		
x3	{ a }	{ b }	Ø	
x4	{ a, b }	{ a }	{ a, b }	Ø

Unterscheid-
barkeitsmatrix

Unterscheidbarkeitsfunktion

- Definiere für jedes Attribut a_i eine boolesche Variable a_i*
- Setze $c_{ij}^* = \left\{ a^* \middle| a \in c_{ij} \right\}$
- Setze $f_S\left(a_1^*, \ldots, a_m^*\right) = \wedge\left\{ \vee c_{ij}^* \middle| 1 \le j \le i \le n, c_{ij} \ne \phi \right\}$
- Die Menge aller **Primimplikanten***) von f_S bestimmt die Menge aller Redukte.

*) *Wiederholung:* Primimplikanten kann man z.B. leicht aus der disjunktiven Minimalform einer Funktion ablesen. Für $f(x,y,z) = x \wedge y \vee y \wedge z$ wären es $x \wedge y$ und $y \wedge z$.

Unterscheidbarbeitsfunktion

	x1	x2	x3	x4
x1	Ø			
x2	{ a, b }	Ø		
x3	{ a }	{ b }	Ø	
x4	{ a, b }	{ a }	{ a, b }	Ø

Unterscheid-
barkeitsmatrix

Unterscheid-
barkeitsfunktion

$$f_S(a,b) = (a \vee b) \wedge (a) \wedge (a \vee b) \wedge (b) \wedge (a) \wedge (a \vee b) = a \wedge b$$

Entscheidungsklassen

- Das Entscheidungsattribut bestimmt eine Partition des Universums:

$$CLASS_S(d) = \left\{ X_S^1, \ldots, X_S^{r(d)} \right\}$$

- $CLASS_S(d)$ heißt **Klassifizierung der Objekte in S, bestimmt durch die Entscheidung d.**
- $r(d)$ ist die Anzahl der Werte, die das Entscheidungsattribut annehmen kann.
- Die Menge X_S^i heißt ***i*-te Entscheidungsklasse von S**.

B-positive Region

- Sind X_1 ... $X_{r(d)}$ die Entscheidungsklassen von S, so heißt

$$POS_B(d) = \underline{B}X_1 \cup \ldots \cup \underline{B}X_{r(d)}$$

B-positive Region von *S*.

- ***Interpretation:*** Die Menge aller Objekte, die sich eindeutig einer Entscheidung zuordnen lassen.

Beispiel: B-positive Region

	Kopf-schmerzen	Muskel-schmerzen	Temperatur	Grippe
e1	ja	ja	normal	nein
e2	ja	ja	hoch	ja
e3	ja	ja	sehr hoch	ja
e4	nein	ja	normal	nein
e5	nein	nein	hoch	nein
e6	nein	ja	sehr hoch	ja

$$POS_{\{K.,M.,T.\}}(d) = \{\, e1, e2, e3, e4, e5, e6 \,\} = U$$

Beispiel: B-positive Region

x	*Temperatur*	*krank*
x1	37 - 38	Nein
x2	38 - 39	Ja
x3	38 - 39	Nein
x4	> 39	Ja

} keine eindeutige Zuordnung!

$$POS_{\{Temp.\}}(d) = \{x1, x4\} \neq U$$

Allgemeine Entscheidung

- Die **allgemeine Entscheidung** in einem Entscheidungssystem S ist eine Abbildung

$$\partial_A : U \to 2^{V_d}$$
$$x \mapsto \left\{ i \,\middle|\, \exists x' \in U : x' \, IND(A) \, x \wedge d(x') = i \right\}$$

- ***Interpretation:*** Die Menge aller Entscheidungswerte, die x und die zu x äquivalenten Objekte haben.

Konsistenz

- Ein Entscheidungsystem heißt **konsistent (deterministisch)**, falls $\left|\partial_A\left(x\right)\right| = 1 \; \forall x \in U$
  Dann gilt: $POS_A(d) = U$

- Es heißt **inkonsistent (indeterministisch)**, falls $\neg\left(\left|\partial_A\left(x\right)\right| = 1 \; \forall x \in U \right) \quad \Leftrightarrow \quad \exists x \in U : \left|\partial_A\left(x\right)\right| \neq 1$
  Dann gilt: $POS_A(d) \neq U$

Algorithmische Konstruktion von Entscheidunsregeln

- Im Folgenden soll ein Algorithmus zur Konstruktion von Entscheidungsregeln vorgestellt werden:

- Bilde eine Matrix $M^d(S) = ((c_{ij}{}^d))$ durch

$$c_{ij}^d = \begin{cases} \phi, \ falls\ d(x_i) = d(x_j) \\ \left\{ a \in A \,\middle|\, a(x_i) \neq a(x_j) \right\}, \ sonst \end{cases}$$

Beachte: $S = (U, A \cup \{d\}), d \notin A$
Also ist d nicht in c_{ij} enthalten!

Algorithmische Konstruktion von Entscheidungsregeln

- $M^d(S)$ heißt **entscheidungs-relative Unterscheidbarkeitsmatrix von S**.
- Konstruiere

$$f_S^{d,k}\left(a_1^*,\ldots,a_m^*\right)=\wedge\left\{\vee c_{ik}^*\;\middle|\;1\leq i\leq n,\, c_{ik}\neq\phi\right\}$$

- f heißt **(k,d)-relative Unterscheidungsfunktion**.
- Aus den **Primimplikanten*** lassen sich die Regeln konstruieren.

*) *Wiederholung:* Primimplikanten kann man z.B. leicht aus der disjunktiven Minimalform einer (booleschen) Funktion ablesen. Für $f(x,y,z)=x\wedge y\vee y\wedge z$ wären es $x\wedge y$ und $y\wedge z$.

Unterscheidbarkeitsmatrix bilden

	Kopf-schmerzen [k]	Temperatur [t]	Grippe
e1	ja	normal	nein
e2	ja	hoch	ja
e3	ja	sehr hoch	ja
e4	nein	normal	nein
e5	nein	hoch	nein
e6	nein	sehr hoch	ja
e7	nein	hoch	ja
e8	nein	sehr hoch	nein

Unterscheidbarkeitsmatrix bilden

	Kopf-schmerzen [k]	Temperatur [t]	Grippe
e1	ja	no…	
e2	ja		
e3	ja		
e4	nein		
e5	nein		
e6	nein		
e7	nein		
e8	nein		

	e1	**e2**	**e3**	**e4**	**e5**	**e6**	**e7**	**e8**
e1	Ø	{t}	{t}	Ø	Ø	{k,t}	{k,t}	Ø
e2	…t}	Ø	Ø	{k,t}	{k}	Ø	Ø	{k,t}
e3	[illegible]	Ø	Ø	{k,t}	{k,t}	Ø	Ø	{k}
e4	Ø	{k,t}	{k,t}	Ø	Ø	{t}	{t}	Ø
e5	Ø	{k}	{k,t}	Ø	Ø	{t}	Ø	Ø
e6	{k,t}	Ø	Ø	{t}	{t}	Ø	Ø	Ø
e7	{k,t}	Ø	Ø	{t}	Ø	Ø	Ø	{t}
e8	Ø	{k,t}	{k}	Ø	Ø	Ø	{t}	Ø

Entscheidungsfunktionen konstruieren

$$r_1 = t \wedge t \wedge (k \vee t) \wedge (k \vee t) = t$$

$$r_2 = \ldots = k \wedge t$$

$$r_3 = k \wedge t$$

$$r_4 = t$$

$$r_5 = k \wedge t$$

$$r_6 = t$$

$$r_7 = t$$

$$r_8 = k \wedge t$$

	e1	e2	e3	e4	e5	e6	e7	e8
e1	Ø	{t}	{t}	Ø	Ø	{k,t}	{k,t}	Ø
e2	{t}	Ø	Ø	{k,t}	{k}	Ø	Ø	{k,t}
e3	{t}	Ø	Ø	{k,t}	{k,t}	Ø	Ø	{k}
e4	Ø	{k,t}	{k,t}	Ø	Ø	{t}	{t}	Ø
e5	Ø	{k}	{k,t}	Ø	Ø	{t}	Ø	Ø
e6	{k,t}	Ø	Ø	{t}	{t}	Ø	Ø	Ø
e7	{k,t}	Ø	Ø	{t}	Ø	Ø	Ø	{t}
e8	Ø	{k,t}	{k}	Ø	Ø	Ø	{t}	Ø

Entscheidungsregeln bilden

	Kopf-schmerzen [k]	Temperatur [t]	Grippe
e1	ja	normal	nein
e2	ja	hoch	ja
e3	ja	sehr hoch	ja
e4	nein	normal	nein
e5	nein	hoch	nein
e6	nein	sehr hoch	ja
e7	nein	hoch	ja
e8	nein	sehr hoch	nein

$$r_1 = t$$
$$r_2 = k \wedge t$$
$$r_3 = k \wedge t$$
$$r_4 = t$$
$$r_5 = k \wedge t$$
$$r_6 = t$$
$$r_7 = t$$
$$r_8 = k \wedge t$$

Entscheidungsregeln bilden

R1: Wenn **Temperatur=normal**, folgt **Gippe=nein**

R2: Wenn **Kopfschmerzen=ja** und **Temperatur=hoch**, folgt **Grippe=ja**

R3: Wenn **Kopfschmerzen=ja** und **Temperatur=sehr hoch**, folgt **Grippe=ja**

R4: Wenn **Temperatur=normal**, folgt **Grippe=nein**

R5: Wenn **Kopfschmerzen=nein** und **Temperatur=hoch**, folgt **Grippe=nein**

R6: Wenn **Temperatur=sehr hoch**, folgt **Grippe=ja**

R7: Wenn **Temperatur=hoch**, folgt **Grippe=ja**

R8: Wenn **Kopfschmerzen=nein** und **Temperatur=sehr hoch**, folgt **Grippe=nein**

Noch etwas zu Entscheidungsregeln

R1: Wenn **Temperatur=normal**, folgt **Gippe=nein**

R2: Wenn **Kopfschmerzen=ja** und **Temperatur=hoch**, folgt **Grippe=ja**

R3: Wenn **Kopfschmerzen=ja** und **Temperatur=sehr hoch**, folgt **Grippe=ja**

R4: Wenn **Temperatur=normal**, folgt **Grippe=nein**

R5: Wenn **Kopfschmerzen=nein** und **Temperatur=hoch**, folgt **Grippe=nein**

R6: Wenn **Temperatur=sehr hoch**, folgt **Grippe=ja**

R7: Wenn **Temperatur=hoch**, folgt **Grippe=ja**

R8: Wenn **Kopfschmerzen=nein** und **Temperatur=sehr hoch**, folgt **Grippe=nein**

- ***R3*** wird durch ***R6*** überflüssig.
- ***R8*** steht mit mit ***R6*** in Konflikt.

Regelanwendung

- Umgang mit Konflikten durch "**Voting**":
  - Bestimme zu jeder zutreffenden Regel den **Support** - die Anzahl der Objekte, die mit dieser Regel im ursprünglichen Entscheidungssystem klassifiziert werden.
  - Gewichte die möglichen Entscheidungen anhand des Regel-Supports und nimm die mit dem höchsten Gewicht.
- Trifft keine Regel zu, nimm die am häufigsten vorkommende Entscheidung.

Erzeugung optimaler Regeln

- Regeln sind oft zu groß und kompliziert.
- Konstruktion möglichst einfacher, robuster. Regeln ist erwünscht.
- Regeln werden einfacher, wenn weniger Attribute zugrunde liegen.

➡ Selektierung von wirklich relevanten Attributen, gekoppelt mit Bestimmung sogenannter **Redukte** ist wichtig!

Redukte

- Bestimmung von Äquivalenzklassen hilft, Speicherplatz zu sparen (nur Repräsentanten müssen gespeichert werden).
- Es geht aber noch besser:

Finde eine Untermenge B' der Attribute B, so dass die Partitionierung von X unter B' erhalten bleibt.

- Überflüssige Attribute werden also eliminiert.

Redukte - ein Beispiel

	Abschluss	Erfahrung	Französich	Referenz	Entscheidung
x1		mittel		exzellent	einstellen
x2	MBA	wenig	ja	neutral	ablehnen
x3		wenig		gut	ablehnen
x4	MSc	hoch	ja	neutral	einstellen
x5		mittel		neutral	ablehnen
x6	MSc	hoch	ja	exzellent	einstellen
x7		hoch		gut	einstellen
x8	MCE	wenig	nein	exzellent	ablehnen

Ermittlung von Redukten

- Leider: NP-hartes Problem.
- Aber: Gute Heuristiken vorhanden.
- Problem kann auf einen Algorithmus zur **Minimierung boolescher Ausdrücke** (ähnlich wie Erzeugung von Entscheidungsregeln) reduziert werden - darum auch NP-hart!

Bestimmung optimaler Redukte

Es gibt viele Ansätze, z.B.:

- Dynamische Redukte über zufällige Teilmengen des Universums
- Bestimmung und Einbeziehung von Abhängigkeiten zwischen Attributen
- Berechnung der Relevanz von Attributen: Ein Attribut ist umso relevanter, je mehr sich die Klassifizierung durch Weglassen des Attributs ändern würde.
- …

Weitere Ansätze in der Rough Set Theory

- Attribute müssen diskret sein, was in der Praxis aber nicht unbedingt der Fall ist (z.B. Puls, Spannung, …)

  ⇒ Diskretisierungsmechanismen für kontinuierliche Werte.

- Rauscheliminierung

Quellen:

- J. Komorowski, Z. Pawlak, L. Polkowski, A. Skowron - Rough Sets: A Tutorial

- Z. Pawlak,J. Grzymala-Busse, R. Slowinski, W. Ziarko - Rough Sets

Das war's.

Fragen?