Vertiefung in die Dempster-Shafer-Theorie

Soft Computing 2
Sommersemester 2010

Fynn Feldpausch

Universität Bremen, Fachbereich 3

20. Mai 2010

Überblick

1 Grundlagen

2 Kombinationsregeln
- Smets Kombinationsregel
- Yagers Kombinationsregel

3 Umgang mit Widersprüchen
- Ermittlung in einem Mordfall
- Verwendung anderer Regeln

4 Anwendungsbeispiel: Szenenanalyse
- Motivation und Systemüberblick
- Aufbereitung der Sensordaten
- Unsichere Daten und belief-Kombination
- Inference by Information Gain (IBIG)

5 Exkurs: Komplexität von Kombinationsregeln

6 Zusammenfassung

Grundlagen

Grundlagen

Frame of Discernment

- Menge Θ sich gegenseitig ausschließender Elemente
- Interpretation der Elemente als Hypothesen, Aussagen

{A} Es existieren Planeten und es existiert Leben.
{B} Es existieren Planeten, aber kein Leben.
{C} Es existieren keine Planeten.
{A,B} Entspricht $A \vee B$, also: Es existieren Planeten.
{Θ} Unwissenheit.

Grundlagen

Frame of Discernment

- Menge Θ sich gegenseitig ausschließender Elemente
- Interpretation der Elemente als Hypothesen, Aussagen
  - {A} Es existieren Planeten und es existiert Leben.
  - {B} Es existieren Planeten, aber kein Leben.
  - {C} Es existieren keine Planeten.
  - {A,B} Entspricht $A \vee B$, also: Es existieren Planeten.
  - {Θ} Unwissenheit.
- Potenzmenge des Betrachtungsrahmens 2^{Θ} ergibt hierarchisch angeordnete Aussageklassen

$$2^{\Theta} = \{\emptyset, \{A\}, \{B\}, \{C\}, \{A, B\}, \{A, C\}, \{B, C\}, \Theta\}$$

Grundlagen

Mass Functions

- Funktion misst die Unterstützung für eine Aussage

Theorem

*Ist Θ ein Betrachtungsrahmen, dann heißt eine Funktion $m : 2^\Theta \rightarrow [0, 1]$ **mass function**, wenn gilt:*

1. $m(\emptyset) = 0$
2. $\sum_{X \subseteq \Theta} m(X) = 1$

- Summe der Werte muss 1 ergeben
- Unwissenheit (*uncommitted belief*) wird Θ zugeordnet

Grundlagen

Belief Functions

- **mass functions** lassen sich mittels bestimmter Regeln kombinieren
- Eine **belief function** misst dann die **gesamte** Unterstützung für eine Aussage

Grundlagen

Belief Functions

- **mass functions** lassen sich mittels bestimmter Regeln kombinieren
- Eine **belief function** misst dann die **gesamte** Unterstützung für eine Aussage

Theorem

Ist Θ ein Betrachtungsrahmen, dann heißt eine Funktion $Bel : 2^{\Theta} \rightarrow [0, 1]$ ***belief function****, wenn gilt:*

1. $Bel(\emptyset) = 0$
2. $Bel(\Theta) = 1$
3. *Für jede natürliche Zahl n und jede Sammlung $X_1, \ldots, X_n$ von Untermengen von Θ,*

$$Bel(X_1 \cup \cdots \cup X_n) \geq \sum_{I \subset \{1,\ldots,n\}, I \neq \emptyset} (-1)^{|I|+1} Bel\left(\bigcap_{i \in I} X_i\right)$$

- Mögliche Funktionen sind:
  - Addition (*Aufsammeln*) aller Teilmengen
  - ausdruckslose Vertrauensfunktion (*vacuous belief function*)

Grundlagen

Upper Probability & Degree of Doubt

- Eine **upper probability function** misst die **maximale** Unterstützung für ein X

Theorem

*Sei **m** eine mass function. Eine Funktion $P^* : 2^\Theta \rightarrow [0, 1]$ heißt **upper probability function**, wenn gilt:*

$$P^*(X) = \sum_{A \cap X \neq \emptyset} m(A)$$

Grundlagen

Upper Probability & Degree of Doubt

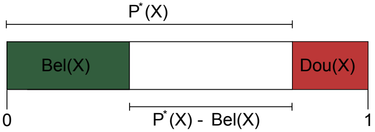
- Eine **upper probability function** misst die **maximale** Unterstützung für ein X

Theorem

*Sei **m** eine mass function. Eine Funktion $P^* : 2^\Theta \to [0,1]$ heißt **upper probability function**, wenn gilt:*

$$P^*(X) = \sum_{A \cap X \neq \emptyset} m(A)$$

- **Darüber hinaus gilt:**
  - $P^*(A) = 1 - Dou(A)$
  - $Dou(A) = Bel(\overline{A})$



Kombinationsregeln

Kombinationsregeln

Dempsters Kombinationsregel

- Regel zur Verrechnung zweier **mass functions**
- Kombination durch:

$$(m_1 \oplus m_2)(X) = \begin{cases} 0, & \text{wenn } X = \emptyset \\ \frac{\sum_{A,B \subseteq \Theta, A \cap B = X} m_1(A) m_2(B)}{1 - \sum_{A,B \subseteq \Theta, A \cap B = \emptyset} m_1(A) m_2(B)}, & \text{sonst} \end{cases}$$

Kombinationsregeln

Dempsters Kombinationsregel

- Regel zur Verrechnung zweier **mass functions**
- Kombination durch:

$$(m_1 \oplus m_2)(X) = \begin{cases} 0, & \text{wenn } X = \emptyset \\ \frac{\sum_{A,B \subseteq \Theta, A \cap B = X} m_1(A) m_2(B)}{1 - \sum_{A,B \subseteq \Theta, A \cap B = \emptyset} m_1(A) m_2(B)}, & \text{sonst} \end{cases}$$

- **Prinzip:** *»Verwerfe alle inkonsistenten Kombinationen und teile den Rest zwischen den konsistenten Kombinationen auf.«*
- Regel widerspricht manchmal unserer Intuition

Kombinationsregeln

Smets Kombinationsregel

- Regel zur Verrechnung zweier **mass functions**
- Kombination durch:

$$(m_1 \oplus m_2)(X) = \begin{cases} 1 - \sum_{A,B \subseteq \Theta, A \cap B = \emptyset} m_1(A) m_2(B), & \text{wenn } X = \emptyset \\ \sum_{A,B \subseteq \Theta, A \cap B = X} m_1(A) m_2(B), & \text{sonst} \end{cases}$$

Kombinationsregeln

Smets Kombinationsregel

- Regel zur Verrechnung zweier **mass functions**
- Kombination durch:

$$(m_1 \oplus m_2)(X) = \begin{cases} 1 - \sum_{A,B \subseteq \Theta, A \cap B = \emptyset} m_1(A) m_2(B), & \text{wenn } X = \emptyset \\ \sum_{A,B \subseteq \Theta, A \cap B = X} m_1(A) m_2(B), & \text{sonst} \end{cases}$$

- **Prinzip:** *»Kombiniere die Werte ohne spätere Normalisierung.«*
- Regel auch bekannt als *non-normalized conjunctive rule of combination*

Kombinationsregeln

Yagers Kombinationsregel

- Regel zur Verrechnung zweier **mass functions**
- Kombination durch:

$$(m_1 \oplus m_2)(X) = \begin{cases} 0, & \text{wenn } X = \emptyset \\ m_1(\Theta)m_2(\Theta) + \sum_{A,B \subseteq \Theta, A \cap B = \emptyset} m_1(A)m_2(B), & \text{wenn } X = \Theta \\ \sum_{A,B \subseteq \Theta, A \cap B = X} m_1(A)m_2(B), & \text{sonst} \end{cases}$$

Kombinationsregeln

Yagers Kombinationsregel

- Regel zur Verrechnung zweier **mass functions**
- Kombination durch:

$$(m_1 \oplus m_2)(X) = \begin{cases} 0, & \text{wenn } X = \emptyset \\ m_1(\Theta)m_2(\Theta) + \sum_{A,B\subseteq\Theta, A\cap B=\emptyset} m_1(A)m_2(B), & \text{wenn } X = \Theta \\ \sum_{A,B\subseteq\Theta, A\cap B=X} m_1(A)m_2(B), & \text{sonst} \end{cases}$$

- **Prinzip:** *»Weise den Rest nicht der leeren Menge sondern viel mehr Θ zu.«*

Kombinationsregeln

Weitere Kombinationsregeln

- **viele weitere Kombinationsmöglichkeiten vorhanden oder denkbar**

- *Dubois-Prade's rule of combination*
- *weighted operators*
- *Transferable Belief Models (TBMs)*
- *disjunctive combinations*
- ...

Umgang mit Widersprüchen

Umgang mit Widersprüchen

Ermittlung in einem Mordfall

- Gegeben ist ein Mordfall mit drei Verdächtigen und zwei Zeugen.
  - **Verdächtige:** Peter, Paul und Mary
  - **Zeugen:** Michael und Simon
- Seien die drei möglichen Urteile:
  - E_1**:** Peter ist der Mörder.
  - E_2**:** Paul ist der Mörder.
  - E_3**:** Mary ist der Mörder.

Umgang mit Widersprüchen

Ermittlung in einem Mordfall

- Gegeben ist ein Mordfall mit drei Verdächtigen und zwei Zeugen.
  - **Verdächtige:** Peter, Paul und Mary
  - **Zeugen:** Michael und Simon
- Seien die drei möglichen Urteile:
  - E_1**:** Peter ist der Mörder.
  - E_2**:** Paul ist der Mörder.
  - E_3**:** Mary ist der Mörder.

- **Michael:** *»Ich bin mir nahezu sicher, dass Peter der Mörder ist.«*
  - $m_1(E_1)$: $0,99$
  - $m_1(E_2)$: $0,01$
  - $m_1(E_3)$: 0

Umgang mit Widersprüchen

Ermittlung in einem Mordfall

- Gegeben ist ein Mordfall mit drei Verdächtigen und zwei Zeugen.
  - **Verdächtige:** Peter, Paul und Mary
  - **Zeugen:** Michael und Simon
- Seien die drei möglichen Urteile:
  - E_1**:** Peter ist der Mörder.
  - E_2**:** Paul ist der Mörder.
  - E_3**:** Mary ist der Mörder.

- **Michael:** *»Ich bin mir nahezu sicher, dass Peter der Mörder ist.«*
  - $m_1(E_1)$: $0,99$
  - $m_1(E_2)$: $0,01$
  - $m_1(E_3)$: 0

- **Simon:** *»Ich bin mir nahezu sicher, dass Mary der Mörder ist.«*
  - $m_2(E_1)$: 0
  - $m_2(E_2)$: $0,01$
  - $m_2(E_3)$: $0,99$

Umgang mit Widersprüchen

Ermittlung in einem Mordfall

- Ergebnis nach dem ersten Kombinationsschritt (Multiplikation):

	$m_1(\{E_1\}) = 0,99$	$m_1(\{E_2\}) = 0,01$	$m_1(\{E_3\}) = 0$
$m_2(\{E_1\}) = 0$			
$m_2(\{E_2\}) = 0,01$			
$m_2(\{E_3\}) = 0,99$			

Umgang mit Widersprüchen

Ermittlung in einem Mordfall

- Ergebnis nach dem ersten Kombinationsschritt (Multiplikation):

	$m_1(\{E_1\}) = 0,99$	$m_1(\{E_2\}) = 0,01$	$m_1(\{E_3\}) = 0$
$m_2(\{E_1\}) = 0$	0	0	0
$m_2(\{E_2\}) = 0,01$	$0,0099$	$0,0001$	0
$m_2(\{E_3\}) = 0,99$	$0,9801$	$0,0099$	0

Umgang mit Widersprüchen

Ermittlung in einem Mordfall

- Ergebnis nach dem ersten Kombinationsschritt (Multiplikation):

	$m_1(\{E_1\}) = 0,99$	$m_1(\{E_2\}) = 0,01$	$m_1(\{E_3\}) = 0$
$m_2(\{E_1\}) = 0$	0	0	0
$m_2(\{E_2\}) = 0,01$	$0,0099$	$0,0001$	0
$m_2(\{E_3\}) = 0,99$	$0,9801$	$0,0099$	0

- Ergebnis nach dem zweiten Kombinationsschritt (Normalisierung):
  - $m(\emptyset) = 0$
  - $m(E_1) = \frac{0}{1-0,9801-2\cdot 0,0099} = 0$
  - $m(E_2) = \frac{0,0001}{1-0,9801-2\cdot 0,0099} = 1$
  - $m(E_3) = \frac{0}{1-0,9801-2\cdot 0,0099} = 0$

Umgang mit Widersprüchen

Ermittlung in einem Mordfall

- Ergebnis nach dem ersten Kombinationsschritt (Multiplikation):

	$m_1(\{E_1\}) = 0,99$	$m_1(\{E_2\}) = 0,01$	$m_1(\{E_3\}) = 0$
$m_2(\{E_1\}) = 0$	0	0	0
$m_2(\{E_2\}) = 0,01$	$0,0099$	$0,0001$	0
$m_2(\{E_3\}) = 0,99$	$0,9801$	$0,0099$	0

- Ergebnis nach dem zweiten Kombinationsschritt (Normalisierung):
  - » $m(\emptyset) = 0$
  - » $m(E_1) = \frac{0}{1-0,9801-2\cdot 0,0099} = 0$
  - » $m(E_2) = \frac{0,0001}{1-0,9801-2\cdot 0,0099} = 1$
  - » $m(E_3) = \frac{0}{1-0,9801-2\cdot 0,0099} = 0$

- **Urteil:** *»Paul ist mit absoluter Sicherheit der Mörder.«*

Umgang mit Widersprüchen

Verwendung anderer Regeln

- **Ergebnis nach Smets Kombinationsregel:**

Umgang mit Widersprüchen

Verwendung anderer Regeln

- **Ergebnis nach Smets Kombinationsregel:**
  - » $m(\emptyset) = 0,9999$
  - » $m(E_1) = 0$
  - » $m(E_2) = 0,0001$
  - » $m(E_3) = 0$
- **Urteil:** *?*

Umgang mit Widersprüchen

Verwendung anderer Regeln

- **Ergebnis nach Smets Kombinationsregel:**
  - $m(\emptyset) = 0,9999$
  - $m(E_1) = 0$
  - $m(E_2) = 0,0001$
  - $m(E_3) = 0$
- **Urteil:** *?*
- **Ergebnis nach Yagers Kombinationsregel:**

Umgang mit Widersprüchen

Verwendung anderer Regeln

- **Ergebnis nach Smets Kombinationsregel:**
  - $m(\emptyset) = 0,9999$
  - $m(E_1) = 0$
  - $m(E_2) = 0,0001$
  - $m(E_3) = 0$
- **Urteil:** *?*

- **Ergebnis nach Yagers Kombinationsregel:**
  - $m(\emptyset) = 0$
  - $m(E_1) = 0$
  - $m(E_2) = 0,0001$
  - $m(E_3) = 0$
  - $m(\Theta) = 0,9999$
- **Urteil:** *?*

Anwendungsbeispiel: Szenenanalyse

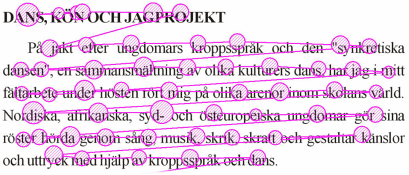
Anwendungsbeispiel: Szenenanalyse

Motivation bioinspirierter Szenenanalyse

- **Ziel:** mit einem Minimum an verfügbaren Ressourcen ein Maximum an Informationen über die Umgebung sammeln

- biologische Systeme haben Optimum für Umgebungsexploration entwickelt
- starke Integration von Verarbeitung der Sensorinformationen (*bottom-up*) und den Informationen der Wissensbasis (*top-down*)

- typisches Beispiel: **Sakkadenbewegungen des menschlichen Auges**
  - » rasante Augenbewegungen ($700^\circ / s$) mit guter Fixierung
  - » effiziente Auswahl »interessanter«Bildausschnitte

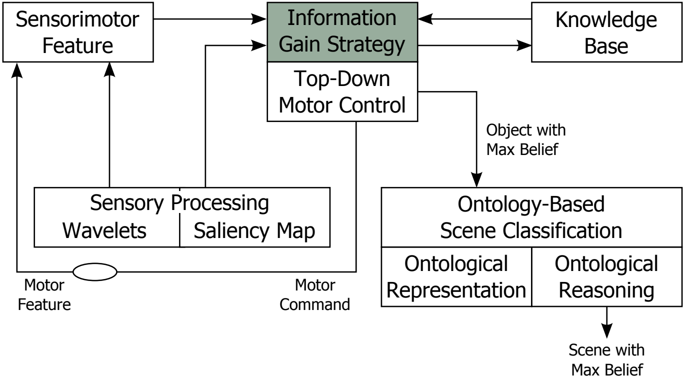


Quelle: http://en.wikipedia.org/wiki/File:ReadingFixationsSaccades.jpg

Anwendungsbeispiel: Szenenanalyse

Systemüberblick

- **dreigliedrige Systemarchitektur:**
  1. *bottom-up feature extraction stage*
  2. *top-down object identification stage*
  3. *knowledge from domain ontology*



Anwendungsbeispiel: Szenenanalyse

Aufbereitung der Sensordaten

- keine Nutzung der Dempster-Shafer-Theorie
- Verwendung von Methoden der Bildverarbeitung

Anwendungsbeispiel: Szenenanalyse

Aufbereitung der Sensordaten

- keine Nutzung der Dempster-Shafer-Theorie
- Verwendung von Methoden der Bildverarbeitung

- Sensorsystem unterstützt **zwei wichtige Funktionen**:
  1. Identifikation von herausragenden Merkmalen
     (mögliche Ziele einer bevorstehenden Fixierung)
  2. Bereitstellung von hochauflösenden Informationen des zur Zeit fixierten Bildausschnittes

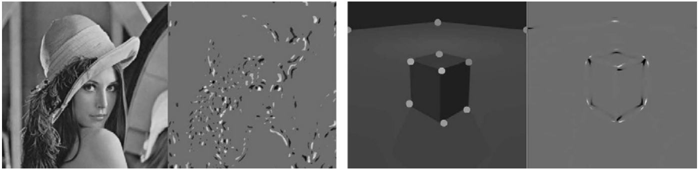
Anwendungsbeispiel: Szenenanalyse

Aufbereitung der Sensordaten

- keine Nutzung der Dempster-Shafer-Theorie
- Verwendung von Methoden der Bildverarbeitung

- Sensorsystem unterstützt **zwei wichtige Funktionen**:
  1. Identifikation von herausragenden Merkmalen (mögliche Ziele einer bevorstehenden Fixierung)
  2. Bereitstellung von hochauflösenden Informationen des zur Zeit fixierten Bildausschnittes

- Resultat 1: **Feature-Karte** (*saliencey map*)



Anwendungsbeispiel: Szenenanalyse

Aufbereitung der Sensordaten

- Resultat 2: **Feature-Vektor** (*sensorimotor feature vector*)
  - » $\overrightarrow{v_1}$ Vektor mit Orientierungsdaten an der fixierten Position
  - » $\overrightarrow{v_2}$ Vektor mit Orientierungsdaten an der fixierten Position
  - » $\overrightarrow{e}$ Bewegungsdaten zwischen Start- und Zielposition
  - » Zusammensetzung zu einem Vektor $\overrightarrow{s} = [\overrightarrow{v_1}, \overrightarrow{e}, \overrightarrow{v_2}]$

Anwendungsbeispiel: Szenenanalyse

Aufbereitung der Sensordaten

- Resultat 2: **Feature-Vektor** (*sensorimotor feature vector*)
  - » $\vec{v_1}$ Vektor mit Orientierungsdaten an der fixierten Position
  - » $\vec{v_2}$ Vektor mit Orientierungsdaten an der fixierten Position
  - » $\vec{e}$ Bewegungsdaten zwischen Start- und Zielposition
  - » Zusammensetzung zu einem Vektor $\vec{s} = [\vec{v_1}, \vec{e}, \vec{v_2}]$

- Mengen von Feature-Vektoren bilden sensomotorische Klassen S_i
- Szene wird im Wesentlichen durch diese S_i repräsentiert

- jedes S_i unterstützt eine Hypothese oder eine Menge von Hypothesen H_k eines Objektes innerhalb der Szene mit einem bestimmten Glaubensmaß $m_i(H_k)$.

Anwendungsbeispiel: Szenenanalyse

Unsichere Daten und *belief*-Kombination

- Kombination verschiedener Hypothesen wird **mittels Dempster-Shafer** durchgeführt

- ist eine Hypothesen-Tupel $(H_k, m(H_k))$ gegeben und es liegt eine Evidenz $(H_l, m_i(H_l))$ aus einer Menge S_i vor, dann wird wie folgt **kombiniert**:

$$\forall H_k \in 2^{\Theta}, H_k \neq \emptyset : m(H_k) = \sum_{H_l, H_n \in 2^{\Theta}, H_l \cup H_n = H_k} m_i(H_l) \cdot m(H_n) \cdot K^{-1}$$

wobei:

$$K = 1 - \sum_{Hl, H_n \in 2^{\Theta}, H_l \cup H_n = \emptyset} m_i(H_l) \cdot m(H_n)$$

Anwendungsbeispiel: Szenenanalyse

Unsichere Daten und *belief*-Kombination

- Kombination verschiedener Hypothesen wird **mittels Dempster-Shafer** durchgeführt

- ist eine Hypothesen-Tupel $(H_k, m(H_k))$ gegeben und es liegt eine Evidenz $(H_l, m_i(H_l))$ aus einer Menge S_i vor, dann wird wie folgt **kombiniert**:

$$\forall H_k \in 2^{\Theta}, H_k \neq \emptyset : m(H_k) = \sum_{H_l, H_n \in 2^{\Theta}, H_l \cup H_n = H_k} m_i(H_l) \cdot m(H_n) \cdot K^{-1}$$

wobei:

$$K = 1 - \sum_{Hl, H_n \in 2^{\Theta}, H_l \cup H_n = \emptyset} m_i(H_l) \cdot m(H_n)$$

(Normalisierungskonstante K ist nicht notwendig; Wegfall gibt Aufschluss über das Konfliktpotential des Systems)

Anwendungsbeispiel: Szenenanalyse

Unsichere Daten und *belief*-Kombination

- aus Komplexitätsgründen wird der Hypothesenraum 2^Θ auf eine hieratchische Struktur T beschränkt, die Objekte (z.B. Stuhl, Tisch, ...) und Objektklassen (z.B. Möbel, ...) der Szene enthält
- *belief*-Werte dieser Objekte bzw. Objektklassen werden vom System erlernt, während Fixierungen vorgenommen werden.
- Wissenbasis wird kontinuierlich erweitert / verbessert

Anwendungsbeispiel: Szenenanalyse

Unsichere Daten und *belief*-Kombination

- aus Komplexitätsgründen wird der Hypothesenraum 2^Θ auf eine hieratchische Struktur T beschränkt, die Objekte (z.B. Stuhl, Tisch, ...) und Objektklassen (z.B. Möbel, ...) der Szene enthält
- *belief*-Werte dieser Objekte bzw. Objektklassen werden vom System erlernt, während Fixierungen vorgenommen werden.
- Wissenbasis wird kontinuierlich erweitert / verbessert
- **Welche Daten bringen den größten Informationsgewinn?**

Anwendungsbeispiel: Szenenanalyse

***Inference by Information Gain* (*IBIG*)**

- **Ziel:** Datum auswählen, das bei Aufnahme zum größten Informationsgewinn bzgl. *belief*-Verteilung im Hypothesenraum führt.
- System identifiziert »interessante« Regionen

Anwendungsbeispiel: Szenenanalyse

***Inference by Information Gain* (*IBIG*)**

- **Ziel:** Datum auswählen, das bei Aufnahme zum größten Informationsgewinn bzgl. *belief*-Verteilung im Hypothesenraum führt.
- System identifiziert »interessante« Regionen

- **Vorgehen:**
  1. kompatible sensomotorische Feature-Vektoren $\hat{S}_i = [\vec{v_1}, \vec{e}, \vec{v_2}]$ identifizieren
     - $\vec{v_1}$ stimmt mit $\vec{v_2}$ der letzten Bewegung überein
     - $\vec{e}$ ist mit der **Feature-Karte** potentieller Ziele vereinbar
  2. für jedes $\hat{S}_i$ alle Hypothesen H_k identifizieren, in denen $\hat{S}_i$ ein $\hat{m}_{T_i}(H_k) \neq 0$ herbeiführt
  3. $\hat{m}_{T_i}(H_k)$ entsprechend der Kombinationsregel ändern
  4. Informationsgewinn berechnen

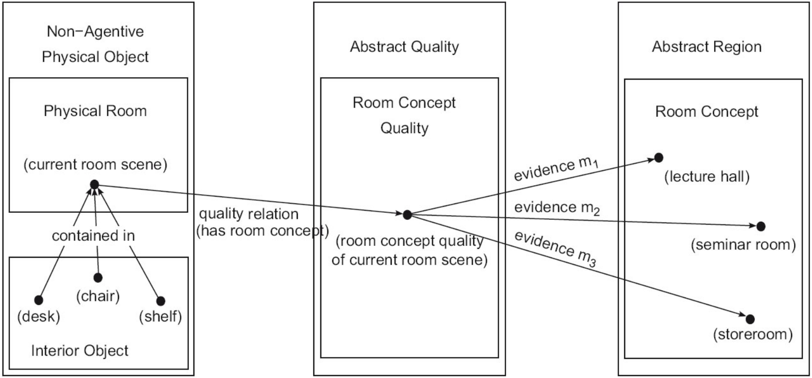
$$\forall \hat{S}_i : I_i(H_k) = |m_T(H_k) - \hat{m}_{T_i}(H_k)|$$

  5. Auswahl treffen

Anwendungsbeispiel: Szenenanalyse

Szenenklassifikation mittels Ontologien

- System muss nicht nur einzelne Objekte behandeln, sondern auch **ganze Räume** identifizieren
- Klassifikation wird mittels **Ontologien** repräsentiert

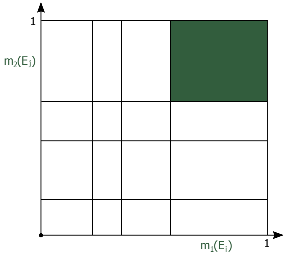


Exkurs: Komplexität von Kombinationsregeln

Komplexität von Kombinationsregeln

Zeitkomplexität bei Kombinationsregeln

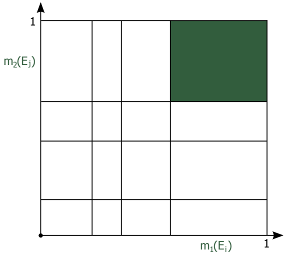
- Hauptproblem bei der Dempster-Shafer-Theorie ist die Zeitkomplexität der Berechnung von Kombinationen
- **alle** oben bereits genannten Regeln besitzen eine **exponentielle Laufzeit (n^m)** bei der Kombination von $m_i(X)$
  - » **n**: Anzahl der verschiedenen Aussagen (E_n)
  - » **m**: Anzahl der Informationen (m_m)



Komplexität von Kombinationsregeln

Zeitkomplexität bei Kombinationsregeln

- Hauptproblem bei der Dempster-Shafer-Theorie ist die Zeitkomplexität der Berechnung von Kombinationen
- **alle** oben bereits genannten Regeln besitzen eine **exponentielle Laufzeit (n^m)** bei der Kombination von $m_i(X)$
  - **n**: Anzahl der verschiedenen Aussagen (E_n)
  - **m**: Anzahl der Informationen (m_m)



- selbst unter der Annahme von **berechenbaren Kombinationen** bleibt die Berechnung von $Bel(X)$ **zu komplex**

Komplexität von Kombinationsregeln

Zeitkomplexität bei Kombinationsregeln

- Für bestimmte Spezialfälle lässt sich $Bel(X)$ in weniger als n^m Schritten berechnen. **Was ist aber mit dem allgemeinen Fall?**
- Es lässt sich zeigen ([ORPONEN]), dass die Berechnung auch dann *#P-complete* ist.

Komplexität von Kombinationsregeln

Zeitkomplexität bei Kombinationsregeln

- Für bestimmte Spezialfälle lässt sich $Bel(X)$ in weniger als n^m Schritten berechnen. **Was ist aber mit dem allgemeinen Fall?**
- Es lässt sich zeigen ([ORPONEN]), dass die Berechnung auch dann *#P-complete* ist.

- **#P-complete**
  - **NP-Problem:** Gibt es eine Variablenbelegung, welche eine gegebene Formel in KNF zu wahr auswerten lässt?
  - **#P-Problem:** Wie viele Variablenbelegung gibt es, welche eine gegebene Formel in KNF zu wahr auswerten lässt?
  - **Ein #p-Problem ist mindestens so schwierig, wie das zugehörige NP-Problem.**
  - **#P-complete-Problem:** Jedes Problem aus **#P** kann auf das betrachtete Problem reduziert werden.

Komplexität von Kombinationsregeln

Zeitkomplexität bei Kombinationsregeln

- Für bestimmte Spezialfälle lässt sich $Bel(X)$ in weniger als n^m Schritten berechnen. **Was ist aber mit dem allgemeinen Fall?**
- Es lässt sich zeigen ([ORPONEN]), dass die Berechnung auch dann *#P-complete* ist.

- **#P-complete**
  - **NP-Problem:** Gibt es eine Variablenbelegung, welche eine gegebene Formel in KNF zu wahr auswerten lässt?
  - **#P-Problem:** Wie viele Variablenbelegung gibt es, welche eine gegebene Formel in KNF zu wahr auswerten lässt?
  - **Ein #p-Problem ist mindestens so schwierig, wie das zugehörige NP-Problem.**
  - **#P-complete-Problem:** Jedes Problem aus **#P** kann auf das betrachtete Problem reduziert werden.

- Da angenommen wird, dass $P \neq NP$ ist, ist das Problem nicht berechenbar.

Komplexität von Kombinationsregeln

Zeitkomplexität bei Kombinationsregeln

- ähnliche Probleme bei der Betrachtung der klassischen Aussagenlogik

- **[GÖDEL]sche Unvollständigkeitssätze:**
  1. Jedes hinreichend mächtige formale System ist entweder widersprüchlich oder unvollständig.
  2. In jedem formalen System der Zahlen, das zumindest eine Theorie der Arithmetik der natürlichen Zahlen enthält, gibt es einen unentscheidbaren Satz, also einen Satz, der nicht beweisbar und dessen Negierung ebenso wenig beweisbar ist.

Komplexität von Kombinationsregeln

Zeitkomplexität bei Kombinationsregeln

- ähnliche Probleme bei der Betrachtung der klassischen Aussagenlogik

- **[GÖDEL]sche Unvollständigkeitssätze:**
  1. Jedes hinreichend mächtige formale System ist entweder widersprüchlich oder unvollständig.
  2. In jedem formalen System der Zahlen, das zumindest eine Theorie der Arithmetik der natürlichen Zahlen enthält, gibt es einen unentscheidbaren Satz, also einen Satz, der nicht beweisbar und dessen Negierung ebenso wenig beweisbar ist.

- Dennoch gibt es effiziente Inferenz-Systeme, die auch in der Praxis Anwendung finden. Das Problem ist also theoretisch unberechenbar, aber praktisch berechenbar.

- **Gibt es solche Methoden auch für die Dempster-Shafer-Theorie?**
  (Unter der Existenzannahme eines effizienten Inferenz-Systems für Aussagen der klassischen Logik)

Komplexität von Kombinationsregeln

Unberechenbarkeit von Kombinationen

- exakte Kombination von *m*-Werten bleibt **unberechenbar**
- selbst bei der Nutzung eines bestehenden Inferenz-Systems

Komplexität von Kombinationsregeln

Unberechenbarkeit von Kombinationen

- exakte Kombination von *m*-Werten bleibt **unberechenbar**
- selbst bei der Nutzung eines bestehenden Inferenz-Systems

Theorem

If an algorithm U computes the beliefs precisely then $t_U(n) \geq c \cdot a^n$ *for some* $a > n$.

- betrachtet wird die Berechnung der **belief function** $bel(X)$
- t_U bezeichnet die maximale Laufzeit des Algorithmus
- n bezeichnet die Länge der Eingabe (Länge von Wissensbasis und Anfrage)
- Wahl der Kombinationsregel spielt keine Rolle

Komplexität von Kombinationsregeln

Unberechenbarkeit von Kombinationen

- initiale *m*-Werte entsprechen unserem Glaubensmaß
- genaue Festlegung dieser Größe ist oft sehr schwer
  - *»Ich bin mit zu 83% sicher«*
  - *»Ich bin mit zu 84% sicher«*

Komplexität von Kombinationsregeln

Unberechenbarkeit von Kombinationen

- initiale *m*-Werte entsprechen unserem Glaubensmaß
- genaue Festlegung dieser Größe ist oft sehr schwer
  - *»Ich bin mit zu 83% sicher«*
  - *»Ich bin mit zu 84% sicher«*

- Berechnung der Kombination mit größerer Präzision als die der Eingabedaten ist sinnlos

Komplexität von Kombinationsregeln

Unberechenbarkeit von Kombinationen

- initiale *m*-Werte entsprechen unserem Glaubensmaß
- genaue Festlegung dieser Größe ist oft sehr schwer
  - *»Ich bin mit zu 83% sicher«*
  - *»Ich bin mit zu 84% sicher«*

- Berechnung der Kombination mit größerer Präzision als die der Eingabedaten ist sinnlos

Theorem

An algorithm U computes the beliefs with precision ϵ if for every [...] query Q it generates a real number $U(Q)$, for which $|U(Q) - bel(Q)| \leq \epsilon$.

If [...] U computes the beliefs with precision $\epsilon < \frac{1}{2}$, then $t_U(n) \geq c \cdot a^n$ for some $a > n$.

- für $\epsilon \geq \frac{1}{2}$ kann immer $\frac{1}{2}$ ausgegeben werden, um die Bedingung zu erfüllen

Komplexität von Kombinationsregeln

Berechenbare Kombinationen

- nicht nur die Festlegung exakter *m*-Werte ist oft schwierig
- oft haben selbst Experten gewissen Zweifel an ihrem Wissen
- es gibt also eine Wahrscheinlichkeit $p_0 > 0$, dass die Expertenaussage **falsch** ist

Komplexität von Kombinationsregeln

Berechenbare Kombinationen

- nicht nur die Festlegung exakter *m*-Werte ist oft schwierig
- oft haben selbst Experten gewissen Zweifel an ihrem Wissen
- es gibt also eine Wahrscheinlichkeit $p_0 > 0$, dass die Expertenaussage **falsch** ist

- basierend auf falschem Wissen können nur flasche Schlüsse gezogen werden
- absolute Korrektheit ist also **nicht erforderlich**

Komplexität von Kombinationsregeln

Berechenbare Kombinationen

- nicht nur die Festlegung exakter *m*-Werte ist oft schwierig
- oft haben selbst Experten gewissen Zweifel an ihrem Wissen
- es gibt also eine Wahrscheinlichkeit $p_0 > 0$, dass die Expertenaussage **falsch** ist

- basierend auf falschem Wissen können nur flasche Schlüsse gezogen werden
- absolute Korrektheit ist also **nicht erforderlich**

- **Forderung:** *»Der Algorithmus darf nicht so oft falsch liegen wie der Mensch.«*

$$p_{err} < p_0$$

Komplexität von Kombinationsregeln

Berechenbare Kombinationen

- Annahme führt zu einer neuen Herangehensweise

Definition

*Assume that positive numbers ϵ and p_0 are given. We say that a statement is **reliably true** if it is true with probability $1 - p_0$ or greater. We say that a probabilistic algorithm U **computes the beliefs** with precision ϵ and reliability $1 - p_0$ if for every [...] query Q it is reliably true that $\left|U(Q) - bel(Q)\right| \leq \epsilon$.*

- **Mit anderen Worten:** $P\left(\left|U(Q) - bel(Q)\right| \leq \epsilon\right) \geq 1 - p_0$

Komplexität von Kombinationsregeln

Berechenbare Kombinationen

- Annahme führt zu einer neuen Herangehensweise

Definition

*Assume that positive numbers ϵ and p_0 are given. We say that a statement is **reliably true** if it is true with probability $1 - p_0$ or greater. We say that a probabilistic algorithm U **computes the beliefs** with precision ϵ and reliability $1 - p_0$ if for every [. . .] query Q it is reliably true that $|U(Q) - bel(Q)| \leq \epsilon$.*

- **Mit anderen Worten:** $P(|U(Q) - bel(Q)| \leq \epsilon) \geq 1 - p_0$

- **Problem für Smets und Yagers Regeln berechenbar** (nicht für Dempsters)

Theorem

*For every $\epsilon < \frac{1}{4}$ and $p_0 < 1$ there exists a **linear-time algorithm** [$t_U(n)$ is bounded by some polynomial of n] that computes beliefs with precision ϵ and reliability $1 - p_0$.*

Komplexität von Kombinationsregeln

Parallelisierung

- **Wie wird auch Dempsters Kombinationsregel berechenbar?**

- Monte-Carlo-Algorithmen lassen sich gut parallelisieren
- viele Schritte können unabhängig voneinander durchgeführt werden
- **theoretisch:** je mehr Prozessoren, desto schneller der Algorithmus
- **praktisch:** Kommunikationsprotokolle und Informationsaustausch wirken der Beschleunigung entgegen und werden irgendwann zu zeitaufwendig

- Architekturen zur Parallelisierung von Monte-Carlo-Algorithmen wurden bereits entworfen ($\approx$ 1990) und effizient genutzt

Zusammenfassung

Zusammenfassung

- **Kombinationsregeln**
  - Dempsters Kombinationregel ist nicht die einzige Regel
  - viele weitere sind denkbar, u.a. Smets, Yagers, ...

Zusammenfassung

- **Kombinationsregeln**
  - Dempsters Kombinationregel ist nicht die einzige Regel
  - viele weitere sind denkbar, u.a. Smets, Yagers, . . .

- **Umgang mit Widersprüchen**
  - Dempsters Regel widerspricht manchmal unserer Intuition
  - widersprüchliche Informationen können zu kuriosen Ergebnissen führen

Zusammenfassung

- **Kombinationsregeln**
  - Dempsters Kombinationregel ist nicht die einzige Regel
  - viele weitere sind denkbar, u.a. Smets, Yagers, ...
- **Umgang mit Widersprüchen**
  - Dempsters Regel widerspricht manchmal unserer Intuition
  - widersprüchliche Informationen können zu kuriosen Ergebnissen führen
- **Anwendungsbeispiel: Szenenanalyse**
  - Nutzung von Feature-Vektoren
  - Kombination von Werten bei der Szenenanalyse
  - Szenenrepräsentation durch Ontologien

Zusammenfassung

- **Kombinationsregeln**
  - Dempsters Kombinationregel ist nicht die einzige Regel
  - viele weitere sind denkbar, u.a. Smets, Yagers, ...
- **Umgang mit Widersprüchen**
  - Dempsters Regel widerspricht manchmal unserer Intuition
  - widersprüchliche Informationen können zu kuriosen Ergebnissen führen
- **Anwendungsbeispiel: Szenenanalyse**
  - Nutzung von Feature-Vektoren
  - Kombination von Werten bei der Szenenanalyse
  - Szenenrepräsentation durch Ontologien
- **Exkurs: Komplexität von Kombinationsregeln**
  - hohe Zeitkomplexität bei der Berechnung von Kombinationen
  - nicht exakte Lösungen sind berechenbar

Literatur

- DANIEL, M. (2006): *Generalization of the classical Combination Rules to DSm Hyper-Power Sets*. Information & Security. An International Journal, Vol. 20, 2006, 50-64.
- GÖDEL, K. (1931): *Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I*. Monatshefte für Mathematik und Physik 38, S.173-198.
- KREINOVICH, V., BERNAT, A., BORRETT, W., MARISCAL, Y., VILLA, E. (1994): *Monte-Carlo methods make Dempster-Shafer formalism feasible*. CS Department, University of Texas, USA.
- ORPONEN, P. (1990): *Dempster's rule of combination is #P-complete*. Artificial Intelligence 44, S. 245-280.
- SCHILL, K., ZETSCHE, C., HOIS, J. (2008): *A belieb-based architecture for scene analysis: From sensorimotor features to knowledge and ontology*.

Vielen Dank für eure Aufmerksamkeit