26:198:722 Expert Systems

- Dempster-Shafer Belief Functions
- Combining Belief Functions
- Types of Belief Functions
- Belief Functions in Expert Systems

Belief Functions

- The standard text for definitions, etc. is, of course:

  Shafer, G. 1976.
  “A Mathematical Theory of Evidence”
  Princeton University Press

Belief Functions

A *belief function* on a frame Θ is a function $\mathrm{Bel}: 2^{\Theta} \to [0, 1]$ such that:

1 $\mathrm{Bel}(\varnothing) = 0$

2 $\mathrm{Bel}(\Theta) = 1$

3 $\mathrm{Bel}(A_1 \cup \ldots \cup A_n) \geq$

$$\sum_i \mathrm{Bel}(A_i) - \sum_{i<j} \mathrm{Bel}(A_i \cap A_j) + \ldots + (-1)^{n+1} \mathrm{Bel}(A_1 \cap \ldots \cap A_n)$$

Plausibility is defined by $\mathrm{Pl}(A) = 1 - \mathrm{Bel}(\sim A)$

Belief Functions

Basic probability assignments are functions $\mathrm{m}: 2^{\Theta} \to [0, 1]$ such that:

1 $\mathrm{m}(\varnothing) = 0$

2 $\sum_{A \subseteq \Theta} \mathrm{m}(A) = 1$

Then we may define $\mathrm{Bel}(A) = \sum_{B \subseteq A} \mathrm{m}(B)$

Belief Functions

- Example:
  - Consider a frame with three possible outcomes $\{a,b,c\}$
  - Suppose we are given the following basic probability assignment:

$$\mathrm{m}(\{a\}) = .1; \mathrm{m}(\{b\}) = .1; \mathrm{m}(\{c\}) = .1;$$

$$\mathrm{m}(\{a,b\}) = .1; \mathrm{m}(\{a,c\}) = .2; \mathrm{m}(\{b,c\}) = .3;$$

$$\mathrm{m}(\{a,b,c\}) = .1$$

Belief Functions

	Bpa	Bel
∅	0	0
{a}	.1	.1
{b}	.1	.1
{c}	.1	.1
{a,b}	.1	.3
{a,c}	.2	.4
{b,c}	.3	.5
{a,b,c}	.1	1

Belief Functions

	Bpa	Bel	Pl
∅	0	0	0
{a}	.1	.1	.5
{b}	.1	.1	.6
{c}	.1	.1	.7
{a,b}	.1	.3	.9
{a,c}	.2	.4	.9
{b,c}	.3	.5	.9
{a,b,c}	.1	1	1

Belief Functions

- Bpas may be recovered from Bel functions using

$$\mathrm{m}(A) = \sum_{B \subseteq A} (-1)^{|A-B|} \mathrm{Bel}(B)$$

Belief Functions

- The *commonality* function is a function $Q:\ 2^{\Theta} \to [0,\ 1]$
  defined by $Q(A) = \sum_{A \subseteq B} m(B)$

- Bpas may be recovered from commonality functions using

$$m(A) = \sum_{A \subseteq B} (-1)^{|B-A|} Q(B)$$

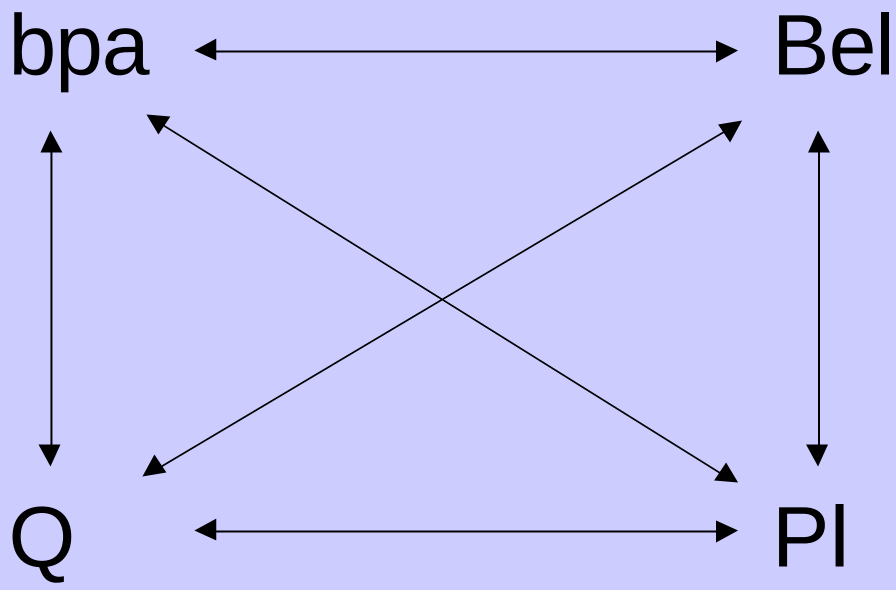
Belief Functions

	Bpa	Bel	Pl	Q
∅	0	0	0	1
{a}	.1	.1	.5	.5
{b}	.1	.1	.6	.6
{c}	.1	.1	.7	.7
{a,b}	.1	.3	.9	.2
{a,c}	.2	.4	.9	.3
{b,c}	.3	.5	.9	.4
{a,b,c}	.1	1	1	.1

Belief Functions

- Recall that the bpa function can be uniquely recovered from Pl, Bel or Q
- In fact, we can convert any one of the four representations uniquely into any of the others
- These conversions are examples of Möbius transforms
- There are Fast Möbius Transforms to do this efficiently (see Kennes paper)

Belief Functions



Belief Functions

- In expert systems based on belief functions:
  - user inputs are often in the form of bpas
  - propagation is most efficient implemented via commonalities
  - marginalization is most efficient implemented via Bel functions
  - output is often desired as Bel or Pl functions

Combining Belief Functions

- Dempster's Rule
  - Consider two belief functions given by their bpas as follows:

$$m_1(\{a\}) = .5; m_1(\{\sim a\}) = .3; m_1(\{a, \sim a\}) = .2;$$

$$m_2(\{a\}) = .7; m_1(\{\sim a\}) = .2; m_1(\{a, \sim a\}) = .1$$

Combining Belief Functions

				m_1		
			$\{a\}$	$\{\sim a\}$	$\{a,\sim a\}$	
			0.5	0.3	0.2	
	$\{a\}$	0.7	0.7x0.5=0.35	0.7x0.3=0.21	0.7x0.2=0.14	
			$\{a\}$	-	$\{a\}$	
m_2	$\{\sim a\}$	0.2	0.2x0.5=0.10	0.2x0.3=0.06	0.2x0.2=0.04	
			-	$\{\sim a\}$	$\{\sim a\}$	
	$\{a,\sim a\}$	0.1	0.1x0.5=0.05	0.1x0.3=0.03	0.1x0.2=0.02	
			$\{a\}$	$\{\sim a\}$	$\{a,\sim a\}$	

$$m_1 \otimes m_2 (\{a\}) = \frac{0.35 + 0.14 + 0.05}{1 - (0.21 + 0.10)} = \frac{0.54}{0.69} = 0.783$$

$$m_1 \otimes m_2 (\{a, \sim a\}) = \frac{0.02}{1 - (0.21 + 0.10)} = 0.029$$

$$m_1 \otimes m_2 (\{\sim a\}) = \frac{0.06 + 0.04 + 0.03}{1 - (0.21 + 0.10)} = \frac{0.13}{0.69} = 0.188$$

Combining Belief Functions

- Note, however, the following:

	m_1	Q_1	m_2	Q_2	$Q_1 x Q_2$	m
{a}	.5	.7	.7	.8	.56	.54
{~a}	.3	.5	.2	.3	.15	.13
{a,~a}	.2	.2	.1	.1	.02	.02

After normalization, these are the same values as derived from Dempster's Rule

Combining Belief Functions

- In expert system applications, therefore, it is efficient to:
  - use Fast Möbius Transforms to convert bpas to commonalities
  - combine the commonalities by pointwise multiplication
  - (eventually) use Fast Möbius Transforms to convert the results back to bpas or other desired outputs

Types of Belief Functions

- If A is a subset of the frame Θ of a belief function, then A is a *focal element* if $\mathrm{m}(A) > 0$
- The *core* of a belief function is the union of all its focal elements
- If, for some subset *A*, $\mathrm{m}(A) = s$ and $\mathrm{m}(\Theta) = 1 - s$ then m is a *simple support function*
- Thus a *simple support function* has only one focal element other than the frame itself

Types of Belief Functions

- A belief function that is the combination of one or more simple support functions is called a *separable support function*
- A belief function that results from marginalizing a separable support function may not itself be separable; it is called a *support function*; Shafer suggests these are fundamental for the representation of evidence

Types of Belief Functions

- Simple support functions

  $\subset$

  Separable support functions

  $\subset$

  Support functions

  $\subset$

  Belief functions
- A belief function whose focal elements are nested is called a consonant belief function

Types of Belief Functions

- A belief function that is not a support function is called a *quasi support function*
- Quasi support functions arise as the limits of sequences of support functions
- A belief function for which $\mathrm{Bel}(A \cup B) = \mathrm{Bel}(A) + \mathrm{Bel}(B)$ whenever $A \cap B = \varnothing$ is called a Bayesian belief function
- Equivalently, a Bayesian belief function is a belief function all of whose focal elements are singletons
- Bayesian belief functions are quasi support functions (except when $\mathrm{Bel}(\{q\}) = 1$ for some $q \in \Theta$)

Belief Functions in Expert Systems

- Belief functions can be propagated locally in Join Trees (Markov Trees) using the Shenoy-Shafer algorithm
- Belief functions can also be propagated locally in Junction Trees using the Aalborg architecture; this requires division (of commonalities) and intermediate results may not be interpretable
- In practice, it is most efficient to perform combination using commonalites and marginalization using Bels

Belief Functions in Expert Systems

- Xu and Kennes give efficient algorithms for carrying out belief function combination, for bit-array representations of subsets, and for Fast Möbius Transforms
- The bit-array representation includes algorithms for testing subsets, forming intersections, unions, etc directly with the bit-arrays
- Full details of the Fast Möbius Transform algorithms are given in Kennes

Belief Functions in Expert Systems

- Efficient implementations are especially important for belief functions
  - n binary variables generate a joint space with 2^n configurations in probability systems
  - n binary variables generate a joint space with 2^{2^n} potential focal elements in belief function systems

Belief Functions in Expert Systems

- "AND" nodes can be defined in belief function terms
  - Suppose we wanted to create a relationship showing that a variable A is true iff variables B and C are both true
  - In a Bayesian network, we could use:

$$\left[\begin{array}{c|c} a,b,c & 1 \\ a,b,\sim c & 0 \\ a,\sim b,c & 0 \\ a,\sim b,\sim c & 0 \\ \sim a,b,c & 0 \\ \sim a,b,\sim c & 1 \\ \sim a,\sim b,c & 1 \\ \sim a,\sim b,\sim c & 1 \end{array}\right]$$

Belief Functions in Expert Systems

- “AND” nodes can be defined in belief function terms
  - Suppose we wanted to create a relationship showing that a variable A is true iff variables B and C are both true
  - What would we use for belief functions?

Belief Functions in Expert Systems

- "AND" nodes can be defined in belief function terms
  - Suppose we wanted to create a relationship showing that a variable A is true iff variables B and C are both true
  - What would we use for belief functions?

$$\left[\left\{(a,b,c),(\sim a,b,\sim c),(\sim a,\sim b,c),(\sim a,\sim b,\sim c)\right\}\middle|1\right]$$

Belief Functions in Expert Systems

- Discounted "AND" nodes can also be defined
  - Suppose we want A to be certain if B and C are both certain, but B and C both to be true with probability 0.95 when A is certain

$$\left[\begin{array}{c|c} a,b,c & 1 \\ a,b,\sim c & 0 \\ a,\sim b,c & 0 \\ a,\sim b,\sim c & 0.0526 \\ \sim a,b,c & 0 \\ \sim a,b,\sim c & 1 \\ \sim a,\sim b,c & 1 \\ \sim a,\sim b,\sim c & 0.9474 \end{array}\right]$$

Belief Functions in Expert Systems

- Discounted "AND" nodes can also be defined
  - Suppose we want A to be certain if B and C are both certain, but B and C both to be true with bpa 0.95 when A is certain

$$\left[\begin{array}{c|c} \{(a,b,c),(\sim a,b,\sim c),(\sim a,\sim b,c),(\sim a,\sim b,\sim c)\} & 0.95 \\ \{(a,b,c),(a,\sim b,\sim c),(\sim a,b,\sim c),(\sim a,\sim b,c),(\sim a,\sim b,\sim c)\} & 0.05 \end{array}\right]$$

Belief Functions in Expert Systems

- Shafer & Srivastava show how to apply mean-per-unit sampling using belief functions
- Gillett & Srivastava show how to perform attribute sampling using belief functions
- Gillett shows how to apply monetary unit sampling using belief functions

Belief Functions in Expert Systems

- Elicitation of bpas from domain experts is potentially more difficult even than for probabilities, partly because of unfamiliarity, but more importantly because far more parameters need to be obtained
- Eliciting expert beliefs in a sufficiently general way that they can be interpreted as either probabilities or bpas for comparative studies is even trickier!

Belief Functions in Expert Systems

- One possibility
  - Elicit two parameters
    - The ratio *f* estimating how much more support the evidence provides for the objective than against it
    - The degree of indeterminacy *i* estimating the extent to which the evidence fails to provide persuasive evidence for or against the objective

Belief Functions in Expert Systems

- One possibility
  - For probabilities

$$\begin{bmatrix} o & \left| \dfrac{f}{1+f} \right. \\ \sim o & \left| \dfrac{1}{1+f} \right. \end{bmatrix}$$

Belief Functions in Expert Systems

- One possibility
  - For belief functions

$$\left[\begin{array}{c|c} o & \dfrac{f-i}{1+f} \\ \sim o & \dfrac{1-f\times i}{1+f} \\ o,\sim o & i \end{array}\right]$$

Belief Functions in Expert Systems

- As in the case of probabilities, joint valuations cannot be uniquely determined from marginals (which is often all domain experts provide)
- Depending on the application, however, "best" or "worst" cases can sometimes be identified

Belief Functions in Expert Systems

- The Shafer & Srivastava paper we read for today sets out extensive arguments why belief functions might be considered superior to probabilities for certain applications, such as auditing
- Among these reasons, the one that first attracted me to study belief functions when I was building an Expert System is the argument that they better represent ignorance
- In auditing, for example, accounts receivable, insufficient replies from customers might lead us to assess a probability of, say, only 70% that accounts receivable exist
- Probability theory then forces us to assess a 30% probability that they do not exist, despite the fact that there is no evidence they do not - merely insufficient evidence that they do

Belief Functions in Expert Systems

- Belief functions allow us to assign a 70% bpa to existence, and the balance to the whole frame, representing ignorance
- In probability theory there would be no difference if *some* of the missing customers in fact wrote to deny the existence of the balance
- Using belief functions, however, we could assign some part of the bpa to represent contrary evidence, and the remainder to ignorance - perhaps $\mathrm{m}(exist) = 0.7; \mathrm{m}(\sim exist) = 0.2; \mathrm{m}(exist, \sim exist) = 0.1$
- Of course, in belief function terms, complete ignorance is represented by $\mathrm{m}(exist, \sim exist) = 1$: it must be one of the outcomes, we don't know which, or which is more likely
- Probabilistically, ignorance is represented as $\mathrm{P}(exist) = \mathrm{P} \sim (exist) = 0.5$ and we have to assume the outcomes equally likely